

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО»**

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

"На правах рукопису"
УДК _____

«До захисту допущено»

Завідувач кафедри

_____ О.В. Коваль

(підпис) (ініціали,

прізвище)

“ ____ ” _____ 2018р.

Магістерська дисертація

зі спеціальності 121 Інженерія програмного забезпечення

за спеціалізацією Програмне забезпечення розподілених систем

на тему “Інтелектуальне діагностування стану силового трансформатора із застосуванням мультиагентного підходу”

Виконав: студент 6 курсу, групи ТВз-71мп

_____ Поліщук Олександр Володимирович

(прізвище, ім'я, по батькові)

(підпис)

Науковий керівник _____ доцент, Ковальчук А.М.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Рецензент _____

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____

(підпис)

Київ - 2018

Національний технічний університет України
“Київський політехнічний інститут ім. Ігоря Сікорського”

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти другий, магістерський

зі спеціальності - 121 Інженерія програмного забезпечення

за спеціалізацією - Програмне забезпечення розподілених систем

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ О.В. Коваль

(підпис)

(ініціали, прізвище)

«_____» _____ 2018р.

ЗАВДАННЯ
на дипломний проект (роботу) студенту
Поліщуку Олександру Володимировичу
(прізвище, ім'я, по-батькові)

1. Тема проекту (роботи): «Інтелектуальне діагностування стану силового трансформатора із застосуванням мультиагентного підходу », науковий керівник: Ковальчук Артем Михайлович, доцент кафедри АПЕПС, затверджені наказом по університету від « 05 » листопада 2018 р. № 4073 - с
2. Строк подання студентом проекту (роботи) : « 11 » грудня 2018 р.
3. Об'єкт дослідження: системи діагностування стану силового трансформатору.
4. Предмет дослідження: методи та засоби інтелектуального діагностування технічного стану та режимів роботи силового трансформатора.
5. Перелік питань, які потрібно розробити
 1. Розробка програмного продукту на основі онтологій із мультиагентною системою;
 2. Здійснення моніторингу та розбору вимог до даних;
 3. Розгляд та аналіз алгоритмів пошуку помилок на основі онтологій;

4. Реалізація пошуку помилок з відповідними даними взятими попередньо із виконаного моніторингу силового трансформатору;
5. Створення зручного інтерфейсу для роботи із програмним продуктом.
6. Орієнтований перелік публікацій
 1. Ковальчук А.М., Поліщук О.В. Інтелектуальне діагностування стану силового трансформатора із застосуванням мультиагентного підходу. зб. наук. праць “ΛΟΓΟΣ. The art of scientific mind” з матеріалами міжнар. наук.-практ. конф., м. Івано-Франківськ, 5 грудня, 2018р. Вінниця : ГО “Європейська наукова платформа”, 2018 Т.4. С. 77-79.
7. Дата видачі завдання « 11 » вересня 2017р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	Строки виконання етапів магістерської дисертації	Примітка
1	Отримання завдання	17.05.2018	
2	Ознайомлення з технічною літературою та збір інформації	01.06.2018-03.09.2018	
3	Дослідження предметної області, існуючих рішень та основних технологій діагностування трансформаторів	03.09.2018-28.09.2018	
4	Аналіз можливостей створення мультиагентних систем та розроблення програмного продукту	01.10.2018-26.10.2018	
5	Захист програмного продукту	22.10.2018	
6	Передзахист	26.11.2018-30.11.2018	
7	Захист	17.12.2018	

Студент _____

Поліщук О.В.

Керівник роботи _____

Ковальчук А.М.

РЕФЕРАТ

Робота виконана на 85 аркушах , містить 28 ілюстрацій, 22 таблиці. При підготовці використовувалась література з 30 джерел.

Актуальність теми зумовлена широким використанням технологій основаних на веб-сервісах, які є важливим фактором, що дають змогу забезпечити вирішення проблем інтелектуального діагностування стану силових трансформаторів на основі онтологій із використанням можливостей отримання даних через онлайн моніторинг.

Мета. Метою даної роботи є розробка системи діагностування несправностей силового трансформатора використовуючи типи системи, що базуються на знаннях онтологій. Вона здатна здійснювати збір даних в реальному часі від компонентів трансформатору та надавати статус компоненту, діагностувати несправності та автоматично будувати графік відповідно до проаналізованих даних.

Завдання. Для досягнення мети потрібно виконати наступні завдання:

- Розробити програмний продукт на основі онтологій із застосуванням мультиагентних систем;
- Здійснити моніторинг та розбір вимог до даних;
- Розглянути та проаналізувати алгоритми пошуку несправностей на основі онтологій;
- Реалізувати пошук несправностей у трансформаторі з відповідними даними взятими попередньо із виконаного моніторингу силового трансформатору;
- Створити зручний інтерфейс для роботи з програмним продуктом.

Об'єкт дослідження. Інтелектуальна діагностика

Предмет дослідження. Діагностування силового трансформатору з застосуванням мультиагентної системи.

Наукова новизна. Наукова новизна полягає в інтелектуальному діагностуванні силового трансформатору на основі онтологій.

Практична цінність. Практична цінність полягає у аналізі інтелектуальних методів та засобів діагностування силових трансформаторів на основі даних взятих із онлайн моніторингу.

Публікації.

Ковальчук А.М., Поліщук О.В. Інтелектуальне діагностування стану силового трансформатора із застосуванням мультиагентного підходу. зб. наук. праць “ΛΟΓΟΣ. The art of scientific mind” з матеріалами міжнар. наук.-практ. конф., м. Івано-Франківськ, 5 грудня, 2018р. Вінниця : ГО “Європейська наукова платформа”, 2018 Т.4. С. 77-79.

Ключові слова. Силовий трансформатор, мультиагентна система, агент, онтологія, алгоритм пошуку помилок.

ABSTRACT

Work carried out on 85 pages containing 28 figures, 22 tables. The paper was written with references to 30 different sources.

Topicality. The urgency of the topic is due to the widespread use of technologies based on web services, which are an important factor in solving the problems of intelligent diagnostics of the status of power transformers on the basis of ontologies using the possibilities of obtaining data through online monitoring.

Purpose. The purpose of this work is to develop a system for diagnosing faults of a power transformer, using types of systems based on the knowledge of ontologies. It is able to collect real-time data from transformer components and provide component status, diagnose malfunctions, and automatically build a graph according to the analyzed data.

Task. To achieve this goal, the following tasks are required:

- develop software based on ontology using multi-agent systems;
- monitor and analyze data requirements;
- consider and analyze algorithms for finding faults on the basis of ontologies;
- perform a fault finding in the transformer with the corresponding data taken from the monitoring of the power transformer;
- create a user-friendly interface for working with a software product.

Object of research. Intelligent diagnostics.

Subject of research. Diagnosis of a power transformer using a multi-agent system.

Scientific novelty. Scientific novelty consists in the intellectual diagnosis of a power transformer based on ontologies.

Practical value of research. The practical value lies in the analysis of intelligent methods and means of diagnosing power transformers based on the data taken with online monitoring.

Publications.

Kovalchuk A.M., Polishchuk A.V. Intelligent diagnostics of power transformer status using multi-agent approach. Scientific journal "ΑΝΓΟΣ. The Art of the Scientific Mind" Ivano-Frankivsk city, December of 5, 2018 year Vinnytsya: NGO "European Science Platform", 2018 №1. T.2. Pp. 14-17.

Keywords. Power transformer, multiagent system, agent, ontology, error search algorithm.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	10
ВСТУП	11
1. СИЛОВИЙ ТРАНСФОРМАТОР ТА МУЛЬТИАГЕНТА СИСТЕМА	13
1.1 Конструкція силового трансформатору.....	13
1.2 Онлайн моніторинг силових трансформаторів	22
1.3 Діагностування несправностей на основі аналізу газу	24
1.4 Передумови автоматизації системи електроживлення	26
1.5. Агент та мультиагентна система.....	28
1.6. Архітектура агента.....	30
1.7. Методологія проектування агентів	31
1.7.1. Високорівнева та середньорівнева модель методології (HLIM).....	32
1.7.2. MaSE методологія.....	32
1.7.3. Gaia методологія	33
1.8. Висновки до розділу	34
2. МУЛЬТИАГЕНТНА СИСТЕМА МОНІТОРИНГУ СИЛОВОГО ТРАНСФОРМАТОРА	35
2.1 Схема моніторингу та діагностування несправностей трансформатора	35
2.2 Система основана на знаннях.....	36
2.3 Діагностика дефектів трансформатора на основі правил	37
2.4 Діагностика дефектів трансформатора на основі онтологій	37
2.5 Взаємодія агентів	38
2.6 Абстрактна модель онтологій	40

2.7 Агент онтологій	46
2.8 Висновки до розділу	49
3. ВИКОРИСТАННЯ ПРОГРАМНОГО ПРОДУКТУ ДЛЯ ІНТЕЛЕКТУАЛЬНОГО ДІАГНОСТУВАННЯ	50
3.1 Опис технології	50
3.2 Засоби реалізації мультиагентної системи	50
3.2.1. Мова програмування: Java	50
3.2.2. SpringBoot	53
3.2.3. Архітектура агентів	54
3.2.4. Мова програмування: JavaScript	55
3.3 Принцип роботи	57
3.4 Висновки	60
4. РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ “ІНТЕЛЕКТУАЛЬНЕ ДІАГНОСТУВАННЯ СИЛОВОГО ТРАНСФОРМАТОРУ”	61
4.1 Опис ідеї проекту	61
4.2 Технологічний аудит ідеї проекту	63
4.3 Аналіз ринкових можливостей	63
4.4 Розробка ринкової стратегії проекту	71
4.5 Розробка маркетингової програми	74
4.6 Висновки	78
ВИСНОВКИ	79
ПЕРЕЛІК ПОСИЛАНЬ	81
ДОДАТОК А	84
ДОДАТОК Б	102

ДОДАТОК В.....	104
----------------	-----

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ІТ	– інформаційні технології
НН	– низька напруга
ВН	– висока напруга
DAU	– data acquisition unit (одиниця збору даних)
УВЧ	– ультрависокочастотна
PD	– partial discharge
ОК	– особистий ключ
LTC	– load tap changer (навантаження перемикача)
AI	– artificial intelligence (штучний інтелект)
DGA	– dissolved gas analysis (аналіз розчиненого газу)
EMS	– energy management system (система енергоменеджменту)
SCADA	– supervisory control and data acquisition (системи контролю доступу до даних)
HMI	– human-machine interface (людсько-машинний інтерфейс)
SAS	– substation automation system (системи автоматизації підстанцій)
DL	– directory facilitator (координатор каталогів)

ВСТУП

Енергетичні системи генерують, передають та розподіляють електричну енергію споживачам. Електрична система розподілу електроенергії є важливою частиною систем електропостачання в постачанні електроенергії споживачам.

Електроенергетичні підприємства в усьому світі все частіше використовують комп'ютерний моніторинг, контроль та управління розподілом електроенергії через різноманітні системи для надання кращих послуг споживачам.

Тому останні дослідження і розвиток ведеться в області інформаційних технологій (ІТ) та передачі даних для автоматизації даного процесу.

Силовий трансформатор є важливим компонентом енергетичної системи, який повинен правильно та надійно функціонувати для надійної роботи системи.

Тому інтелектуальне діагностування стану силового дає можливість захистити систему від серйозних проблем, контролювати роботу трансформатору в онлайн режимі.

Під час експлуатації силових трансформаторів виникають різні типи стресів, пов'язаних з навантаженням, температурою навколишнього середовища, часом роботи і т. д. що впливають на його роботу в результаті чого може призвести до втрати електроенергії.

В зв'язку з цим важливо діагностувати помилку якнайшвидше наскільки це можливо, коли виникають певні симптоми. Автоматичні дії можна виконувати навіть в деяких критичних ситуаціях.

Метою даної магістерської дисертації є розробка програмного забезпечення для інтелектуального діагностування стану силового трансформатору.

Розроблений програмний продукт спрямований на забезпечення діагностування стану трансформатору на основі даних із онлайн моніторингу.

Система призначена для представлення в реальному часі інформації про стан силового трансформатору перед розробкою якої було проведено різноманітні науково-дослідні роботи в області діагностики дефектів силового трансформатора,

однак декілька проблем як і раніше залишається невирішеним. Тому ця робота мотивувалася пошуком рішень для наступних проблем:

- Велика кількість даних, отриманих від кожного компонента енергосистеми, вимагає зусилля та знання фахівців у різних сферах;
- Відсутність інтелектуальної гнучкості в існуючих методах діагностування через що вони не можуть бути легко змінені відповідно до вимог ситуації;
- База знань, застосовувана на сьогоднішній день, здатна забезпечити лише інформацією про енергетичну систему в реальний час і не в змозі вибрати відповідну дію в критичні ситуації.

Цілі цієї роботи спрямовані на пошук вирішення проблем, які були описані вище виділені в наступних розділах дисертації.

1. СИЛОВИЙ ТРАНСФОРМАТОР ТА МУЛЬТИАГЕНТА СИСТЕМА

1.1 Конструкція силового трансформатору

Силовий трансформатор є однією з найважливіших частин електричної енергосистеми; він перетворює напругу на вищий або нижчий рівень. Залежно від розміру, їх можна розділити на три групи, як зазначено нижче:

- Малі силові трансформатори, 500 — 7500 кВА;
- Середні силові трансформатори, 7500 — 100 МВА;
- Великі силові трансформатори, 100 МВА і більше.

Очікується, що час роботи силових трансформаторів з моменту введення в експлуатацію до моменту виходу зі строю складає приблизно 30 років, проте експлуатація, що перевищує її рейтингові значення, може призвести до значного скорочення життя.

Силовий трансформатор високої напруги - це пристрій, який складається з великої кількості конструктивних елементів, основними з яких є: магнітна система (магнітопровід), обмотки, ізоляція, виводи, бак, охолоджувальний пристрій, механізм регулювання напруги, захисні та вимірювальні пристрої, візок.

У трансформаторах невеликої потужності бак має верхню знімну кришку, тому під час ремонту необхідно зняти цю кришку, а потім підняти активну частину з бака.

Якщо маса активної частини перевищує 25т, вона встановлюється на донну частину бака, а потім накривається дзвоникоподібною верхньою частиною бака і заливається маслом. Такі трансформатори з нижнім рознімним дном не потребують вантажопідйомних пристроїв, щоб вийняти активну частину, тому що після зливання масла верхня частина бака піднімається, відкриваючи доступ до обмоток і магнітопровода [1].

Для зменшення втрат від потоків розсіювання сталі баки екрануються з внутрішньої сторони пакетами з електротехнічної сталі або пластинами з немагнітних матеріалів (мідь, алюміній).

У магнітній системі наявний магнітний потік трансформатора (звідси назва "магнітопровід"). Магнітопровід є конструктивною і механічною основою трансформатора. Слід зазначити, що якість електротехнічної сталі впливає на допустиму магнітну індукцію та втрати в магнітопроводі.

Магнітопровід і його конструктивні деталі є основою трансформатора, на якій встановлюють обмотки і кріплять провідники, що з'єднують обмотки з вводами, створюючи активну частину [2].

Магнітопровід з насадженими на його стержні обмотками - це активна частина трансформатора. Інші елементи трансформатора називаються неактивними (допоміжними) частинами.

Магнітопровід в трансформаторі виконує дві функції: по-перше, він утворює магнітне коло, по якому замикається основний магнітний потік трансформатора, а по-друге, є основою для встановлення та кріплення обмоток, виводів, перемикачів.

Магнітопровід має шихтовану конструкцію, тобто він виготовляється з тонких (товщиною близько 0,5мм) штампованих пластин з електротехнічної сталі або залізонікелевих сплавів, покритих ізолюючою плівкою (наприклад лаком).

Така конструкція магнітопровода зменшує вихрові струми, які індукуються в ньому змінним магнітним потоком, і тим самим, зменшує втрати енергії в трансформаторі. Використовують також феритові магнітопроводи (осердя) з двох однакових половинок. Стержневі сердечники застосовують в трансформаторах потужністю понад 1кВт.

Силові трансформатори залежно від конструкції магнітопровода бувають трьох типів: стержневі, броньові, бронестержневі, а також тороїдальні.

У трансформаторах з тороїдальними магнітопроводами найбільш повно використовуються магнітні властивості матеріалу. Тороїдальні трансформатори мають малі потужності розсіювання і створюють слабкі зовнішні магнітні поля.

Магнітопровід у тороїдальних трансформаторів має форму кільця, що виготовлене із сталеві стрічки. Ці трансформатори мають меншу масу, оскільки для їхнього виготовлення потрібно менше сталі та провода для обмоток. У них також майже відсутнє магнітне розсіювання. Індукція в тороїдальних осердях більша ніж у броньових або стержневих, тому можна зменшити розміри і вагу трансформаторів.

Трансформатори з тороїдальним осердям мають кращі умови для охолодження обмоток, тому що витки розміщені по всьому тороїду.

При цьому зменшується довжина витка, відповідно витрачається менше міді та підвищується коефіцієнт корисної дії трансформатора.

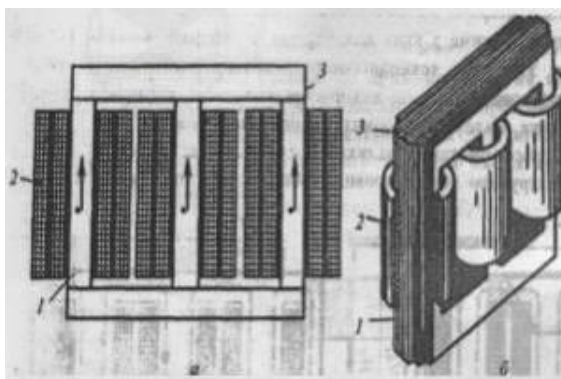


Рисунок 1.1. Магнітопровід трифазного трансформатора стержневого типу з обмотками

У магнітопроводі стержневого типу (рис.1.1, а) вертикальні стержні 1, на яких розміщені обмотки 2, зверху і знизу замкнені ярмами 3. На кожному стержні розміщені обмотки відповідної фази, де проходить магнітний потік цієї фази: у крайніх стержнях - потоки Φ_A і Φ_B , а в середньому стержні - потік Φ_C . На рис.1.1, показано зовнішній вигляд магнітопровіда. При цьому стержні мають ступінчатий переріз, що вписується в круг діаметром d (рис.1.2).

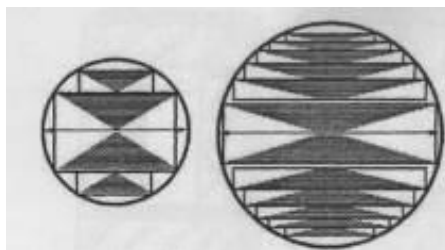


Рисунок 1.2. Форма перерізу стержнів

Стержні трансформаторів великої потужності мають багато ступенів, що забезпечує краще використання площі круга всередині обмотки. Для поліпшення тепловіддачі інколи між окремими пакетами стержня залишають повітряні зазори 5-6 мм, що служать вентиляційними каналами [3].

Магнітопровід броньового типу - це розгалужена конструкція з стержнем та ярмами, які частково прикривають обмотки (бронюють). Магнітний потік у стержні магнітопровода броньового типу в два рази більший, ніж в ярмах, кожне з яких має переріз, у два рази менший від перерізу стержня.

Унаслідок технологічних труднощів виготовлення магнітопроводів броньового типу, їх використовують лише у силових трансформаторах дуже малої потужності (радіотрансформатори).

У трансформаторах великої потужності застосовують бронестержневу конструкцію магнітопровода, яка хоч і вимагає збільшених витрат електротехнічної сталі, але дає змогу зменшити висоту магнітопровода ($H_{\text{бс}} < H_{\text{е}}$) і висоту трансформатора.

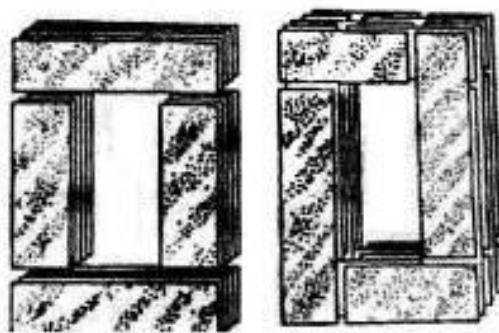


Рисунок 1.3. Стикова (1) і шихтована (2) конструкції магнітопроводів

За способом спряження стержнів з ярмами розрізняють стикову і шихтовану конструкцію стержневого магнітопровода (рис.1.3).

При стиковій конструкції (рис.1.3, 1) стержні і ярма збирають окремо.

Обмотки накладають на стержні, а після цього прикладають верхнє і нижнє ярма, підклавши перед цим ізолюючі прокладки між стикуючими елементами, які зменшують вихрові струми, що виникають при взаємному перекритті листів стержнів і ярм.

Після встановлення двох ярем всю конструкцію пресують і стягують вертикальними шпильками.

Шихтована конструкція магнітопроводів силових трансформаторів показана на (рис.1.3, 2), в якій стержні та ярма збирають на перепліт. Зазвичай шар складається з 2-3 листів.

У наш час магнітопроводи силових трансформаторів виготовляють з холоднокатаної електротехнічної сталі, в якій магнітні властивості вздовж напрямку прокатки листів кращі ніж уперек. Тому при шихтованій конструкції в місцях повороту листів на 90° появляются "зони несуміщення" напрямку прокатки з напрямом магнітного потоку.

На цих ділянках спостерігається збільшення магнітного опору і зростання магнітних втрат. Для послаблення цього явища використовують для шихтовки пластини зі скошеними краями. У цьому випадку замість прямого стику (рис.1.4, а) одержують косий стик (рис.1.4, б), у якого "зона несуміщення" набагато менша.

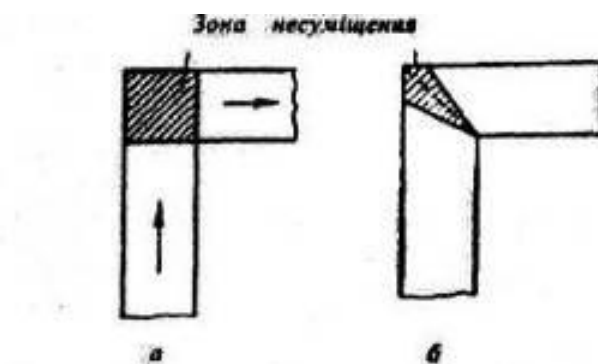


Рисунок 1.4. Зони несуміщення при прямому (а) і косому (б) стиках

Недоліком магнітопроводів шихтованої конструкції є деяка складність збирання, оскільки для насадження обмоток на стержні потрібно розшихтовувати верхнє ярмо, а після насадження обмоток знов його зашихтовувати.

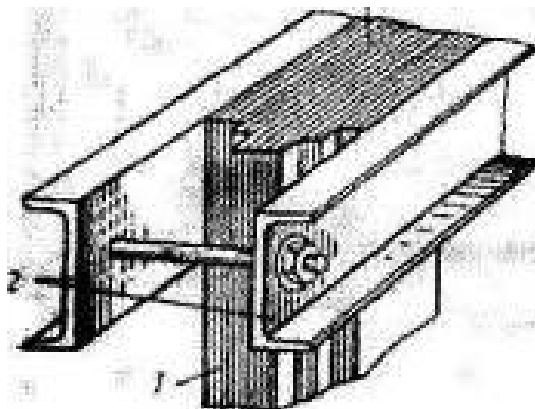


Рисунок 1.5. Опресування ярма:

1 — стержні; 2 — ярмові балки

Стержні магнітопроводів для запобігання розхитуванню обпресовують (закріплюють), накладаючи на стержні бандаж із склострічки або сталевих дротів.

Для обпресування ярма (рис.1.5) та місць їх спряження з стержнями 1 використовують ярмові балки 2, які в місцях, що виходять за крайні стержні стягують шпильками.

Щоб запобігти виникненню різниці потенціалів між металевими частинами під час роботи трансформатора, що може викликати пробій ізоляційних проміжків, які розділяють ці частини, Магнітопровід і деталі його кріплення обов'язково заземлюють.

Заземлення виконують мідяними стрічками, які вставляють між сталевими пластинами магнітопровода одними кінцями та прикріплюють їх до ярмових балок другими кінцями.

У сучасних трансформаторах для обмотки використовують транспонований провід, в якому окремі провідники в паралельному пучку періодично змінюють своє положення [4].

При цьому вирівнюється опір елементарних провідників, збільшується механічна міцність, зменшується товщина ізоляції та розміри магнітопровода.

Обмотки трансформаторів повинні мати достатню електричну та механічну міцність. Ізоляція обмоток і відводів від неї мають бути без пошкоджень, витримувати комутаційні й атмосферні перенапруги.

Обмотки повинні витримувати електродинамічні зусилля, які з'являються при протіканні струмів КЗ (короткого замикання). Необхідно передбачити надійну систему охолодження обмоток, щоб не виникав недопустимий перегрів ізоляції.

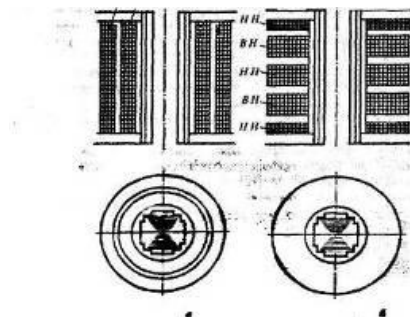


Рисунок 1.6. Обмотки трансформаторів (а) концентричні; (б) дискові

За взаємним розміщенням на стержні обмотки трансформаторів можуть бути концентричними та дисковими.

У першому випадку обмотки низької напруги *НН* і високої напруги *ВН* виконують у вигляді циліндрів і розташовують на стержні концентрично одна відносно іншої (рис.1.6, а).

Така конструкція прийнята в більшості силових трансформаторів. У другому випадку обмотки *ВН* і *НН* виконуються у вигляді невисоких циліндрів з однаковими діаметрами і розташовують на стержні одна над іншою (рис.1.6, б).

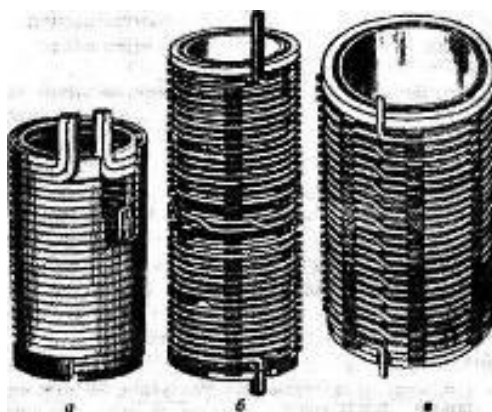


Рисунок 1.7. Конструкція концентричних обмоток

Для того, щоб всі паралельні проводи мали однакове струмове навантаження, виконують транспозицію (перекладку) цих проводів.

У трансформаторах з масляним охолодженням магнітопровід з обмотками розміщується у баку з трансформаторним маслом (рис.1.7).

Трансформаторне масло відбирає тепло від обмоток і магнітопровода, які нагрілись внаслідок проходження по них струму. Трансформаторне масло, володіючи більш високою теплопровідністю ніж повітря, через стінки бака і труби радіатора віддає тепло в навколишнє середовище.

Наявність трансформаторного масла забезпечує більш надійну роботу високовольтних трансформаторів, оскільки електрична міцність масла набагато більша від повітря. Масляне охолодження інтенсивніше від повітряного, тому габарити та вага масляних трансформаторів менші, ніж сухих трансформаторів такої ж потужності.

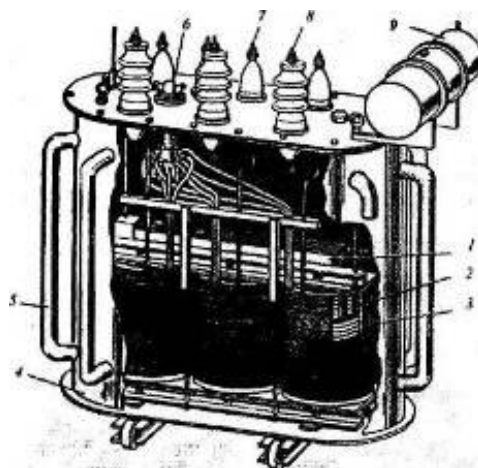


Рисунок 1.8. Конструкція трансформатора з масляним охолодженням

У трансформаторах потужністю до 20-30 кВ/А використовують баки з гладкими стінками. Щоб збільшити поверхню потужніших трансформаторів, яка охолоджується, стінки бака виготовляють ребристими або використовують трубчасті баки.

Масло нагрівається, піднімається догори, а охолоджуючись, опускається вниз. При цьому масло циркулює в трубах, що сприяє більш швидкому його охолодженню.

Для компенсації об'єму масла при зміні температури, а також для захисту масла від окислення і зволоження при контакті з повітрям в трансформаторах використовують розширювальний бачок 9, що є циліндричною посудиною, яка розміщується над кришкою бака і сполучається з ним трубами.

Коливання рівня масла із зміною його температури відбувається не в баку, який завжди заповнений маслом, а в розширювальному бачку, який сполучений через вивід з атмосферою.

Трансформатор має дві або три обмотки, які розміщені на спільному магнітопроводі (осерді), що виготовлений з феромагнітного матеріалу (рис.1.8).

Магнітопровід трансформатора виготовляють завжди замкнутим, щоб магнітний потік проходив по осерді і не розсіювався в повітрі. Змінний магнітний потік індукє у вторинній обмотці змінну ЕРС, яка залежить від кількості витків цієї обмотки та швидкості зміни магнітного потоку згідно з законом електромагнітної індукції [5].

Широко використовуються броньові трансформатори, які мають розгалужений магнітний потік (рис.1.9). У них магнітопровід з середнім більш широким стержнем і двома боковими стержнями меншої ширини нагадують букву Ш, що замикається додатковою пластинкою. Обмотки розміщують на котушці, що натягується на середній стержень. Найчастіше використовують циліндричні обмотки.

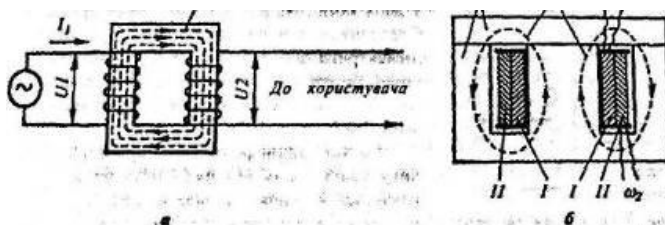


Рисунок 1.9. Принцип роботи трансформатора
Основні конструкційні вузли трансформаторів зображені на рис.1.10.

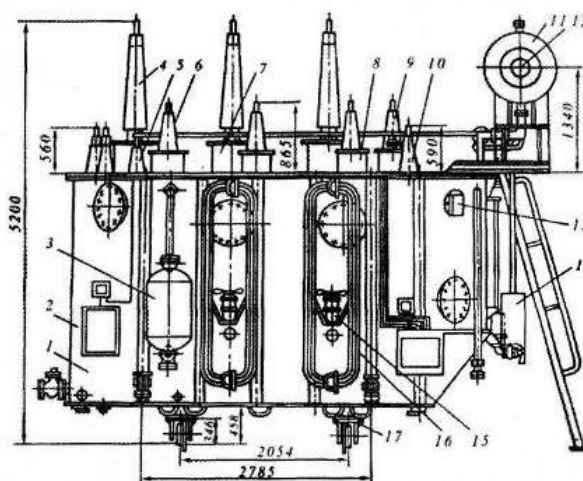


Рисунок 1.10. Трансформатор трифазний триобмотковий

1.2 Онлайн моніторинг силових трансформаторів

Онлайн моніторинг силового трансформатора - це процес доступу до даних в той час як трансформатор працює. Характеристики компонентного онлайн-моніторингу можуть бути різними, це залежить від кількості моніторингових параметрів та доступності необхідних даних.

Для системи онлайн моніторингу зазвичай дані, звіти та сигнали треба періодично фіксувати. Для моніторингу функцій електростанції та пристроїв необхідні наступні основні компоненти:

- Датчики: знімають інформацію або дані про обладнання;
- Одиниці збору даних (DAU): вимірюють та збирають сигнали від різних датчиків;
- Комунікаційна лінія між DAU і комп'ютером: різні види мережевого зв'язку для передачі даних від DAU та датчиків до відповідного програмного забезпечення;
- Комп'ютер із попередньо доступним програмним забезпеченням: програмна платформа для моніторингу компонентів та зв'язку з контрольними об'єктами.

Онлайн моніторинг значно підвищує ефективність роботи та обслуговування процедури силового трансформатора [6]. Характеристики онлайн-моніторингу можуть варіюватися; вони залежать від кількості контрольованих параметрів та доступності потрібних даних.

Статус компонентів силового трансформатора можуть бути зафіксовані різними типами електронних датчиків, у тому числі надвисокочастотними (УВЧ).

Основними параметрами онлайн моніторингу силового трансформатора є:

- Аналіз розчиненого газу в нафті: це один з найефективніших інструментів діагностики для визначення проблем у роботі трансформатора. Перегрів, частковий розряд або локальний розрив викликають наявність декількох газів,

розчинених у маслі. Таким чином, ідентифікація наявності цих газів допомагає вказувати на несправності.

- Вологість в маслі: Наявність вологи в маслі спричиняє зменшення діелектричної сила з наступним зниженням міцності ізоляції. Важливо виміряти рівень вологості до будь-якої несправності.
- Часткова розрядження (PD): PD - тип помилки в силовий трансформатор, що часто відбувається у випадку діелектричного пробою. Рівень PD можна виміряти за допомогою різних методів, таких як електричні та акустичні, що надають інформацію про зміни в силовому трансформаторі.
- Температура масла: температура масла є одним з основних параметрів у загальних температурних умовах силового трансформатора, який включає в себе температуру навколишнього середовища, верхню температуру масла, роботу вентилятора та навантаження. Ці фактори важливі для визначення стану трансформатора під час його експлуатації.
- Температура обмотки: один з граничних факторів для можливості завантаження.. Механічна міцність паперової ізоляції в силовому трансформаторі може бути зменшена за рахунок тривалого впливу надмірного тепла.
- Напруга навантаження та напруга живлення: Автоматичне відстеження струму навантаження та напруги живлення трансформатору збільшить їх термін служби, обмеживши їх максимальне навантаження.
- Коефіцієнт потужності ізоляції: вся електрична ізоляція має вимірну кількість діелектричних втрат незалежно від стану. Хімічні речовини та вологість можуть збільшити втрати.
- Робота насоса / вентилятора: робота вентилятора призначена для регулювання температури трансформатора в різних умовах. Його ненормальна робота може спричинити провал системи охолодження. Статус вентилятора та насоса

можна зафіксувати, вимірюючи їх струм та його співвідношення з вимірюваною температурою.

- Робота натискного перемикача (LTC): несправності в LTC або механічні, або електричні. Невдачі можуть бути викликані поганим дизайном або невідповідністю контактів, високого навантаження, надмірного числа змінних.

Таким чином, системи онлайн моніторингу забезпечують детальну інформацію про стан компонентів силового трансформатора та за допомогою відповідного програмного забезпечення мінімізувати ймовірність несподіваного відключення [7].

1.3 Діагностування несправностей на основі аналізу газу

Проблеми в трансформаторі можуть виникати з дефектів / недоліків і збільшувати свій розвиток в негативному плані для трансформатора. Підвищення температури, вологості, кисню та інших ключових факторів трансформатора під час роботи можуть істотно сприяти деградації ізоляції.

Згідно з типовими несправностями, які пов'язані з високою напругою в трансформаторі, як показано на рис. 1.11.

Як показано, втулки та обмотки мають найвищий ризик невдачі, тому що ці регіони працюють під найвищим електропостачанням. У цьому випадку для визначення несправності компонентів силових трансформаторів необхідні методи діагностики несправностей.

Різні методи діагностики, наприклад хімічні, електричні, теплові, оптичні та механічні можуть бути застосовані в режимі онлайн та оффлайн, щоб діагностувати несправність трансформатора [8].

Ці методи можуть бути застосовані з точки зору різних методів, таких як штучний інтелект, нечітка логіка, машинне навчання.

Наприклад, в методах АІ дані з конкретного інструменту (наприклад, зразки газу) є зібрані в експертній системі для полегшення прийняття рішень з точки зору виявлення несправностей.

Для випадку діагностики несправностей, що базується на даних температури, теплова модель застосовує можливість прогнозування температури трансформаторів у різних місцях.

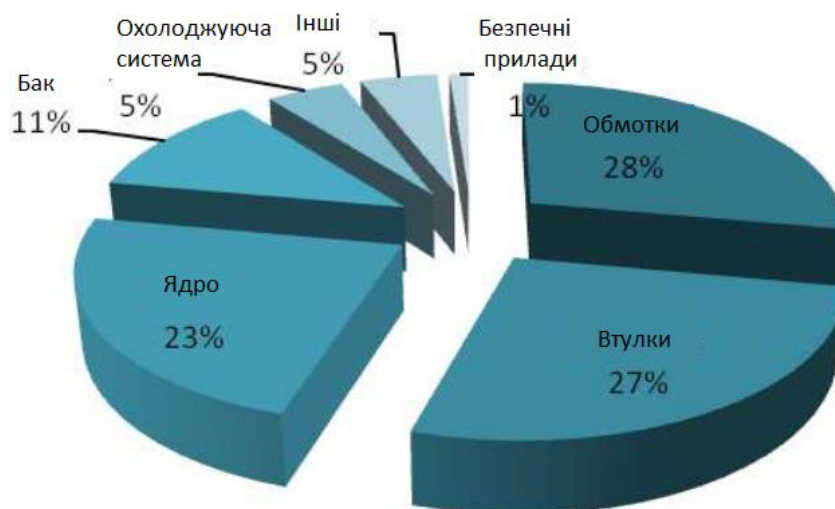


Рисунок 1.11. Розподіл несправностей трансформатора високої напруги

DGA (Dissolved Gas Analysis) є загальноприйнятою методикою діагностики, яка використовувалася протягом декількох десятиліть для діагностування стану масляних трансформаторів.

Зразки масла взяті з трансформатора направляються на діагностику несправностей, такі як гарячі плями, перегрів, частковий розряд дуг і т.д.

В результаті перегрівання масла можуть утворитися вуглеводневі фрагменти та розкладання водню на молекули вуглеводню при електричній та термічній напрузі [9].

Ключові гази такі як: ацетилен (C_2H_2), метан (CH_4), водень (H_2), етилен (C_2H_4), етан (C_2H_6), монооксид вуглецю (CO) і т. д. можуть утворюватися шляхом комбінування фрагментів.

Наприклад, несправність перегріву може генерувати наступні гази: $H_2 \Rightarrow CH_4 \Rightarrow C_2H_6 \Rightarrow C_2H_4 \Rightarrow C_2H_2$.

Діагностика несправностей DGA базується на ключових газах та співвідношенні газів (таких як $R1 = CH_4 / H_2$, $R2 = C_2H_2 / C_2H_4$, $R3 = C_2H_2 / CH_4$, $R4 = C_2H_6 / C_2H_2$, $R5 = C_2H_4 / C_2H_6$) шляхом застосування різних методів виявлення несправностей.

Кожен метод використовує різні коефіцієнти газу (або ключові гази) для класифікації несправностей. Очевидно, що більше категорій класифікації дають кращі результати діагностики. У оцінці статусу трансформатора потужності, швидкому збільшенню основних газів слід приділяти більше уваги, ніж загальному обсягу газу.

1.4 Передумови автоматизації системи електроживлення

Енергетичні системи генерують, передають та розподіляють електричну енергію споживачам. Комп'ютери застосовуються для ефективного моніторингу, контролю та автоматизації потужності системи та її компонентів [10]. Три типи систем автоматизації, що використовуються в передачі поділяться наступним чином:

- Системи енергоменеджменту (EMS): система, яка запускається за допомогою комп'ютерів для управління, моніторингу та оптимізації роботи енергосистеми;
- Системи контролю та контролю доступу до даних (SCADA): Відповідальні за збір даних, процес події та людсько-машинний інтерфейс (HMI);
- Системи автоматизації підстанцій (SAS): зменшення експлуатації на заводах, допомога моніторингу стану компонентів на одній підстанції

На рисунку 1.12 показана структура системи автоматизації електроживлення. Ієрархічна структура формується окремим SAS кожної підстанції на найнижчому рівні. SCADA система перевершує SAS, а EMS знаходиться на вищому рівні.

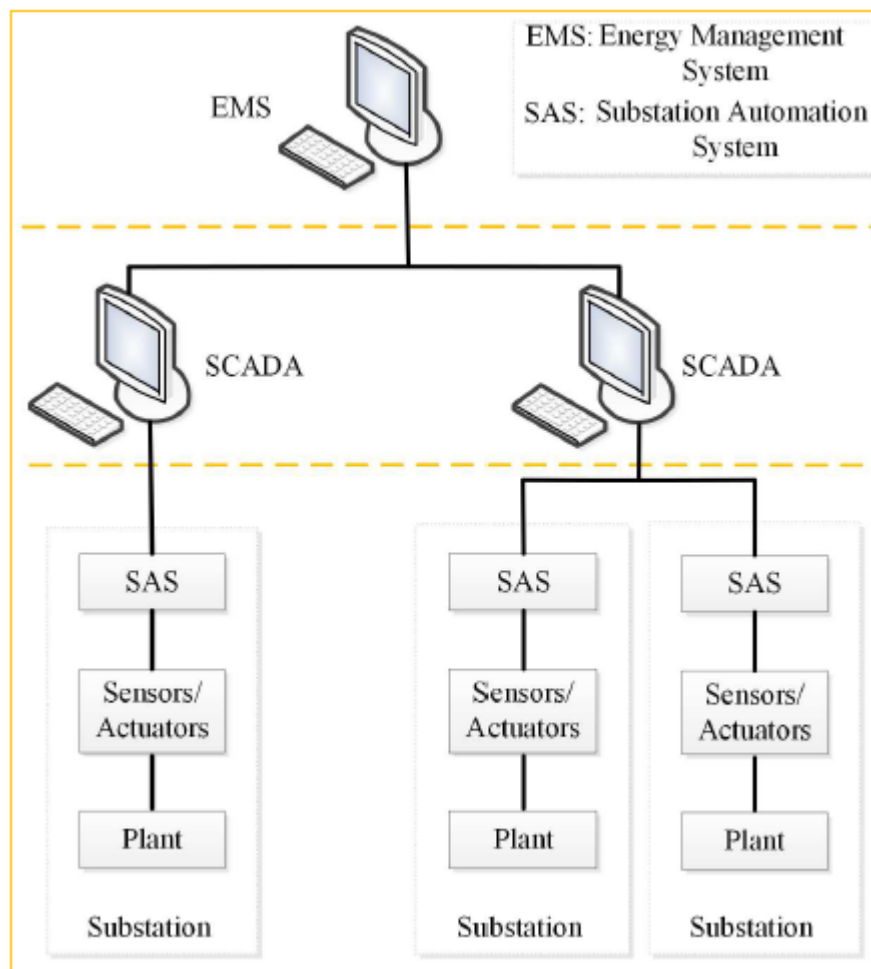


Рисунок 1.12. Автоматизація електроенергетичної системи

SAS використовує безліч пристроїв, інтегрованих у пакет функцій на комунікаційних технологій для моніторингу та управління підстанцією [11].

Насправді, компоненти SAS класифікуються на три рівні: обладнання енергетичної системи, такі як трансформатори та автоматичні вимикачі, які знаходяться на найнижчому рівні, Intelligent Electronic Devices (IEDs) з різними можливостями додатків знаходяться на другому рівні та утиліти підприємства, які знаходяться на найвищому рівні.

IEDs - це мікропроцесорні системи з можливістю відправлення та отримання даних, які розроблені та інтегровані для виконання різних завдань, таких як реле захисту, обстеження навантаження, показники різних компонентів, що передають дані через мережу.

Той факт, що вони розроблені різними постачальниками з різними протоколами можуть спричинити проблеми на стадії спілкування. У цьому випадку рекомендованими є відкриті системи, на основі використання непатентованого стандартного програмного та апаратного інтерфейсу.

Моніторинг є основним рівнем над контролем обладнання в автоматизації енергосистеми. Цей рівень допомагає контролювати стан системи, управляти пристроями комутації та керувати віддалено, і захистити систему від несправностей.

На цьому етапі застосовуються різні методи для онлайн моніторингу стану, щоб забезпечити його функціональні можливості в режимі реального часу, відповідно до вимог.

Інформація фіксується та передається в центр управління, який буде передавати дані на інтелектуальну систему діагностування стану силового трансформатору [12].

Одним з останніх прикладних прийомів моніторингу та контролю елементів електричної енергії являється багатоагентна система.

1.5. Агент та мультиагентна система

Агенти-орієнтоване програмування (АОР) - це ще один тип програмування, парадигма з можливостями інтерфейсів та обміну повідомленнями.

Головна мета агента полягає в тому, щоб виявити автономну сутність, виконати завдання автономно. Основною проблемою об'єктів є те, що вони є пасивними постачальниками послуг і не володіють цією здатністю безпосередньо.

Агенти та програмне забезпечення агента являють собою нову, надійну та корисну техніку здійснену на обчисленнях.

Немає єдиного визначення агента, але найпоширенішого опису набуває визначення, що агентом є комп'ютерна система, розташована в середовищі, і вона здатна діяти автономно від імені користувача або власника.

В цілому агент приймає інформацію з навколишнього середовища своїми датчиками і вирішує, яку дію прийняти для виконання. Дія може бути виконана з виконавцями агента, щоб зробити деякі зміни в його середовищі.

Рисунок 1.13 показує агент з його датчиками / виконавчими елементами, які здатні змінювати середовище.

У складних системах агент не в змозі діяти самостійно контролювати його оточення, тому в цьому випадку може бути використано кілька агентів [13].

Ці агенти мають різні здібності та співпрацюють для досягнення своїх цілей. Таким чином, для досягнення результатів у цьому випадку використовуються мультиагентні системи.

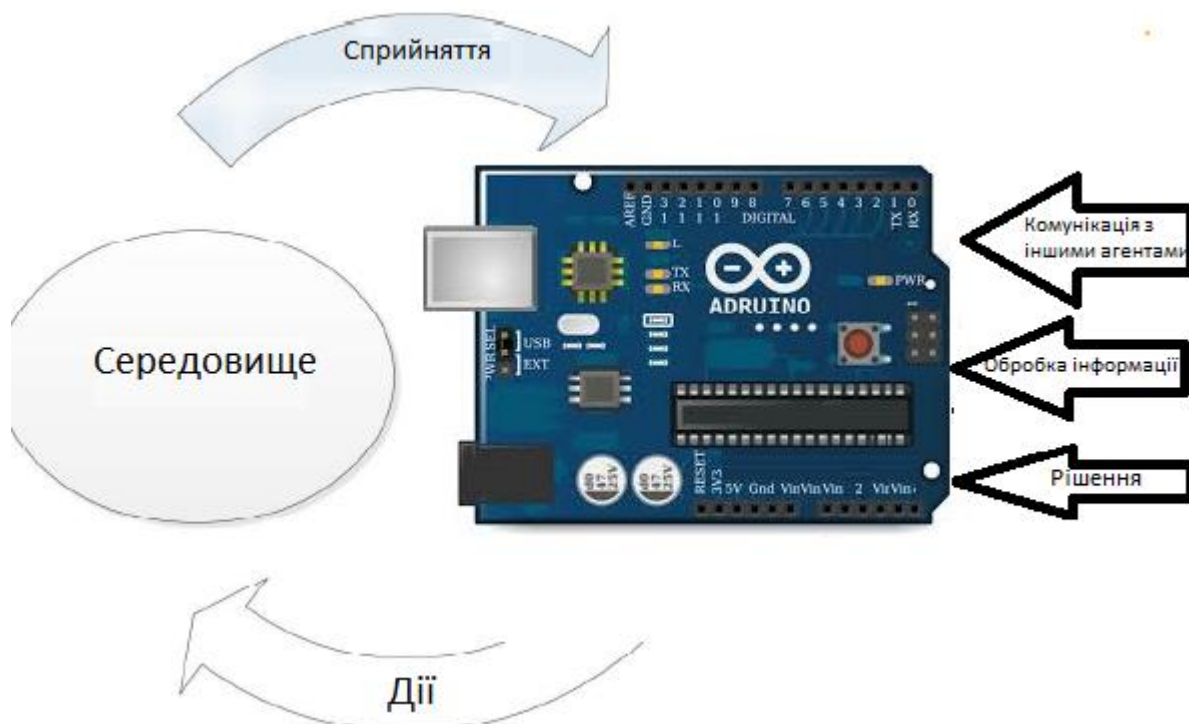


Рисунок 1.13. Середовище агента

Властивості агентів заохочують дослідників застосувати цю технологію для різних сфер. Деякі з цих властивостей підрозділяються наступним чином:

- Автономний: агент може контролювати свої дії без прямого втручання людини;

- Реактивність: агент може сприймати середовище своїми датчиками та реагувати на нього шляхом зміни навколишнього середовища;
- Проактивний: поведінка агента та його відповіді спрямовані виставленими цілями;
- Соціальні можливості: спілкування між агентами, що забезпечує взаємодію один з одним.

Відповідно до властивостей агента і здатності застосовувати різні методи при прийнятті рішення агенти діють дуже схожими на людину.

1.6. Архітектура агента

Різні визначення архітектур агентів надаються дослідниками. Одне визначення надане Майклом Лауком полягає в тому, що: "Архітектури надають інформацію про основні структури даних, відносини між ними, процес або функції, які працюють на цих структурах даних, і цикл операцій виконання агента".

Поняття архітектури агента полягає в тому, як агенти працюють разом для досягнення вирішення складних задач, використовуючи різні парадигми. Вулдрідж і Дженнінгс класифікували архітектури агента в три категорії наступним чином:

- Символічна архітектура;
- Реактивні архітектура;
- Гібридні архітектура.

Основна ідея архітектури символічного міркування полягає в тому, що символи представляють собою навколишнє середовище. Фактично вся інформація про навколишнє середовище написана у формі символів [14].

У цьому випадку основним моментом є опис поточного стану цього навколишнє середовище. Тому ці типи архітектури стикаються з двома ключовими проблемами; перша проблема - це складність перекладу навколишнього середовища в символи.

Друга проблема - це складність реального представлення світу в загальному. Кілька моделей були надані дослідниками для подолання цих проблем.

Реактивна архітектура, розроблена Бруксом і Маесом, є ще одним типом агента архітектура. У цій архітектурі немає символічної або логічної моделі середовища.

Ключовою ідеєю цієї архітектури є те, що в ній може бути сформована інтелектуальна поведінка без явних уявлень і абстрактних міркувань, наданих символічним методом розвідки; Інтелект є невід'ємною властивістю деяких складних систем в яких датчики налаштовані таким чином, що можуть передавати інформацію в реальному часі у відповідні онлайн системи моніторингу.

Найнижчий рівень має менший контроль, ніж вищий рівень стека в ієрархії поведінки, тому прийняття рішень здійснюється шляхом цілеспрямованої поведінки.

Перевага цієї архітектури - це швидша реакція, однак вона працює лише для деяких середовищ.

Гібридна архітектура складається з реактивних і дорадчих архітектур, які діють відповідно до її потреби. Реактивна архітектура корисна в ситуаціях, коли агенти здатні реагувати в часі. Для ситуацій, які вимагають дії агента обгрунтовано, доцільно, то символічна архітектура може бути корисною.

1.7. Методологія проектування агентів

Ключовим моментом розробки MAS є визначення співпраці агентів. Зазвичай методологія охоплює весь життєвий цикл розвитку системи, такий як аналіз, проектування, впровадження та перевірка. На етапі аналізу агенти асоціюються з суб'єктами аналізованого сценарію

Визначаються можливості та відповідальність кожного агента та після цього кроку, нарешті, взаємодія між агентами застосовується [15].

Різні типи методологій були введені дослідниками для різних програмних продуктів. Деякі з запропонованих методик: високорівневі та середні моделі для агенто-орієнтованої методології, MASE та Gaia методології.

На підставі їх здібностей, кожна методологія має деякі переваги та недоліки. Є основи для порівняння цих методологій та їх придатності для конкретних програм.

1.7.1. Високорівнева та середньорівнева модель методології (HLIM)

Дана методологія являє собою розробку систем агентів через серію абстрактних рівнів, де людина, з допомогою машини, може маніпулювати абстракціями на одному рівні всередині абстракцій на наступному нижчому рівні.

В цілому, методологія має дві фази, а саме — етап відкриття і етап визначення. Фаза відкриття спрямовує виявлення агентів та їх поведінку на високому рівні, тоді як етап визначення виробляє реалізовані визначення. HLIM містить п'ять моделей, представлених нижче:

- Високорівнева модель;
- Внутрішня модель агента для внутрішньої структури агента;
- Модель відношення;
- Модель спілкування;
- Модель контракту.

Модель високого рівня забезпечує високий рівень перегляду системи та вихідну точку для розвитку сценарію програми, який відслідковується для опису функціональної поведінки, виявлення агенцій та поведінки в дорозі.

Цілі, плани та вірування аспектів агентів описуються внутрішньою моделлю, яка виводиться безпосередньо з моделі високого рівня [16]. Модель відносин описує взаємозв'язок між агентами, такими як залежність та юрисдикція.

Координація між агентами описується розмовною моделлю. Структура зобов'язань агентів визначається контрактною моделлю. Контракти створюються під час оформлення чи виконання агентів, якщо це необхідно. Ця методологія не має детального опису етапів впровадження та випробувань.

1.7.2. MaSE методологія

Методологія MaSE зосереджена головним чином на робототехніці; він фіксує цілі і продовжує концептуальний етап проектування системи. Кількість моделей на основі графіки використовується для опису типів агента та їх інтерфейсу з іншими агентами, а також незалежне від архітектури визначення внутрішнього дизайну агента.

Методика MaSE складається з аналізу та компонування дизайну. Аналітична частина надає огляд необхідної системи, а конструкторська частина моделює його в корисну конструкцію для впровадження MAS.

Частина аналізу фіксується в три етапи: захоплення цілей, застосування випадків використання, і вдосконалення ролей. У першому кроці виконуються вимоги користувачів, цілі фіксуються і перетворюються на цілі системи рівня.

З цього рівня використовуються випадки та послідовності графіків, які витягуються для ініціалізації множини ролей та шляхів їх зв'язку. Нарешті, третім кроком є вдосконалення та розширення початкового набору ролей та визначення завдань для виконання кожної мета у кроці очищення ролей [17]. Створення класів агента, побудова розмов та дизайн системи - це три етапи проектування.

В методології MaSE зв'язок між агентами застосовується через кінцевий стан машини, що веде до алгебраїчного опису розмов. З цієї причини метод вважається успішним і прийнятним для опису взаємодій між агентами.

1.7.3. Gaia методологія

Методологія Gaia передбачає полегшення аналітика поступово змінюватися від початкового неоднозначного стану до більш короткого та методичного дизайну, який може бути реалізований безпосередньо.

Відповідно до методології Gaia, процес побудови MAS схожий на процес організаційного проектування. Отже, взаємозв'язок та взаємодія між ієрархічними ролями в одній організації буде визначатися як аналітична категорія цієї методології Gaia, яка, у свою чергу, складається з двох інших підкатегорій, а саме моделі та моделі взаємодії.

Наприклад, рольовий приклад у цьому випадку складається з ролі "Data Collector", і його обов'язок полягає в тому, щоб збирати дані з будь-яких повідомлень, отриманих від пристроїв або обладнання на заводі, і зберігати їх у відповідних базах даних та таблицях [18].

Згідно Gaia, кожна роль має різні дозволи та обов'язки, пов'язані їх завданням. Кожна роль повинна відповідати своїм завданням, щоб вона була виконана правильно.

Дозвіл визначає інформаційні ресурси, призначені для кожної ролі. Ці ресурси можуть бути знаннями або інформацією, пов'язаними з агентом. Тому деякі ресурси можуть бути виконані за допомогою ролі, визначеної як життєздатність. З іншого боку, іноді агент не змінює існуючого стану. Це потрібно для певної мети, і це визначається як безпека.

1.8. Висновки до розділу

У цьому розділі описано силовий трансформатор та мультиагентну систему, а саме:

- Розглянуто конструкцію силового трансформатора. Описано його складові та їхню роботу. Також було проаналізовано процес доступу до даних трансформатора під час його роботи - онлайн моніторинг. За допомогою якого фіксується статус компонента за допомогою відповідних датчиків.

- Проаналізовано передумови автоматизації системи електроживлення, а саме: системи енергоменеджменту, системи контролю доступу до даних та системи автоматизації підстанцій.
- Розглянуто середовище агента та що являється його головною метою в роботі системи діагностування. Також було описано архітектуру агента та їхню методологію проектування.

2. МУЛЬТИАГЕНТНА СИСТЕМА МОНІТОРИНГУ СИЛОВОГО ТРАНСФОРМАТОРА

2.1 Схема моніторингу та діагностування несправностей трансформатора

Загальна структура розробленої мультіагентної системи представлена на рисунку 2.1.

Як видно на рисунку 2.1 структура ієрархії складається з трьох рівнів; компоненти на найнижчому рівні, агенти знаходяться на другому рівні, і, нарешті, верхній рівень складається з користувачів і систем знань для процесів моніторингу та прийняття рішень, відмічених як моніторинг та рівень прийняття рішень.

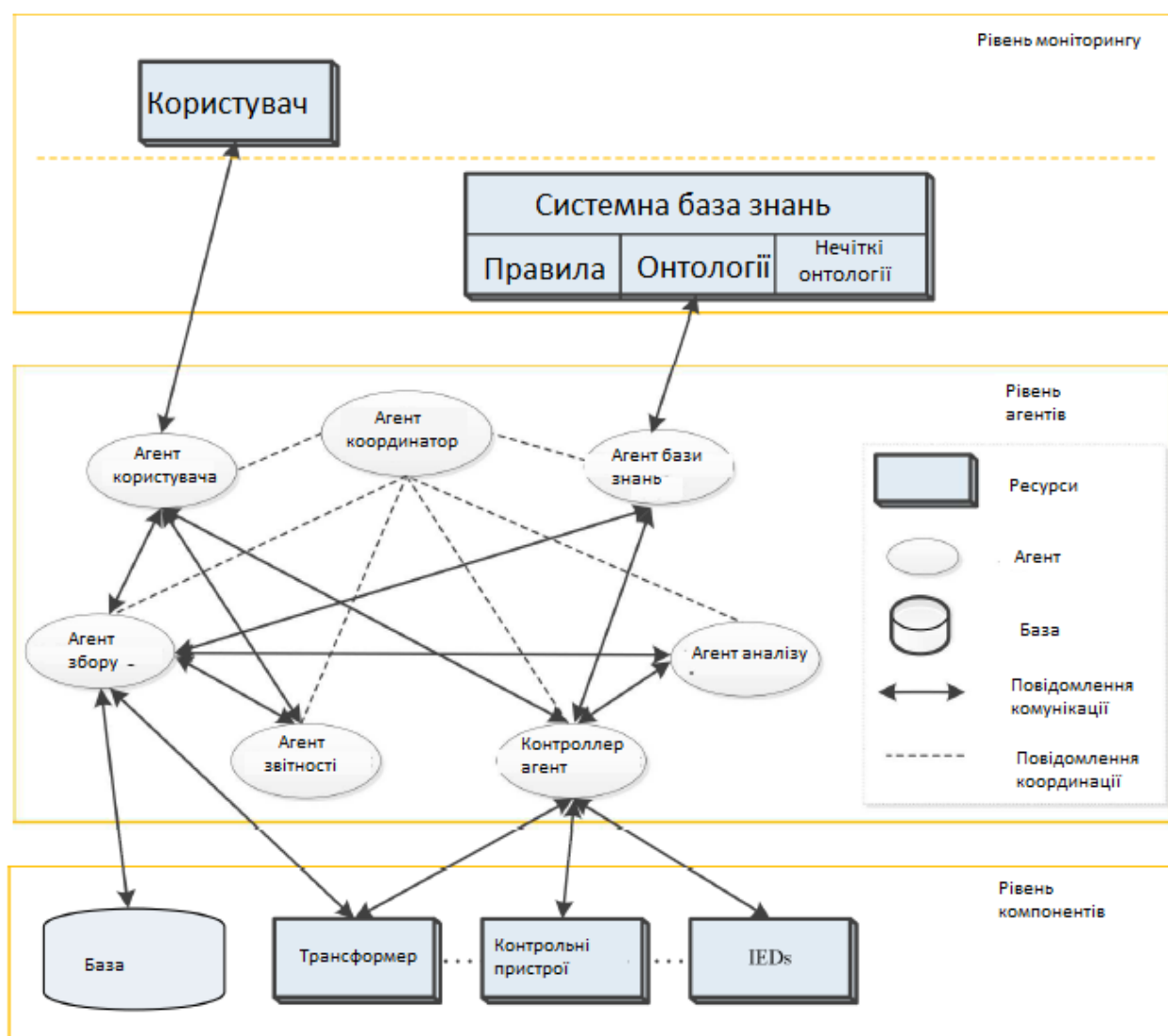


Рисунок 2.1. Структура мультиагентної системи

Тим не менш, пріоритетом користувача є ключова зацікавленість впливу на загальну систему [19]. Рівень компонентів містить обладнання, таке як трансформатор, автоматичні вимикачі, DAU, бази даних, IED і т. д., Які поєднуються з різними агентами для вилучення даних, експлуатації, моніторингу стану, виявлення несправностей тощо.

Рівень агентів представляє різні типи агентів, які аналізуються та розробляються відповідно до методології Gaia. Зазвичай на рівні рішень процес моніторингу та виконання рішень здійснюються лише користувачем.

Розроблена система вводить три типи систем знань на цей рівень, щоб зменшити зусилля користувачів.

2.2 Система основана на знаннях

Система основана на знаннях - це програмна система, яка може імітувати продуктивність людських експертів у обмеженому сенсі. Знання можна представити по-різному: такі як семантична мережа, логіка, процедури, виробничі системи (правила), кадри тощо.

Кожна форма забезпечує різні характеристики. Мета представлення знань полягає у вирішенні проблем, що виникають при інтеграції певного знання в комп'ютерну систему. Це призводить до автоматичного та інтелектуального обґрунтування. У випадку діагностики несправностей розробляються три типи системи знань з різними здібностями [20].

З точки зору цієї роботи, міркування на основі правил було застосовано як першу систему, основану на знаннях, для представлення інформації про діагностику дефектів трансформатора і виконання необхідних дій.

Онтологія була обрана як другий спосіб подання знань для виконання тих самих дій з точки зору онтології. Покращення використання онтологій з точки зору

запропонованої бази знань було досягнуто за допомогою використання нечіткої онтології як третього етапу діагностики дефектів трансформатора на основі DGA.

2.3 Діагностика дефектів трансформатора на основі правил

У запропонованій мультиагентній системі була створена система діагностики несправностей силового трансформатора з міркуваннями на основі правил для зменшення зусиль користувача.

У цій системі досліджуються два додатки правильних міркувань для діагностики несправностей силового трансформатора. Перше застосування - це діагностика силового трансформатора на основі даних DGA.

Стан трансформатора можна діагностувати, виходячи з коефіцієнтів газу та методу Роджера. Іншим способом являється застосування міркування на основі правил від імені користувача в критичній ситуації [21].

Крім того, зниження зусиль користувача може призвести до зниження витрат. Різні агенти реалізуються за допомогою платформи JADE; правила написані в Java.

Рисунок 2.2. демонструє використання у випадку цієї системи.

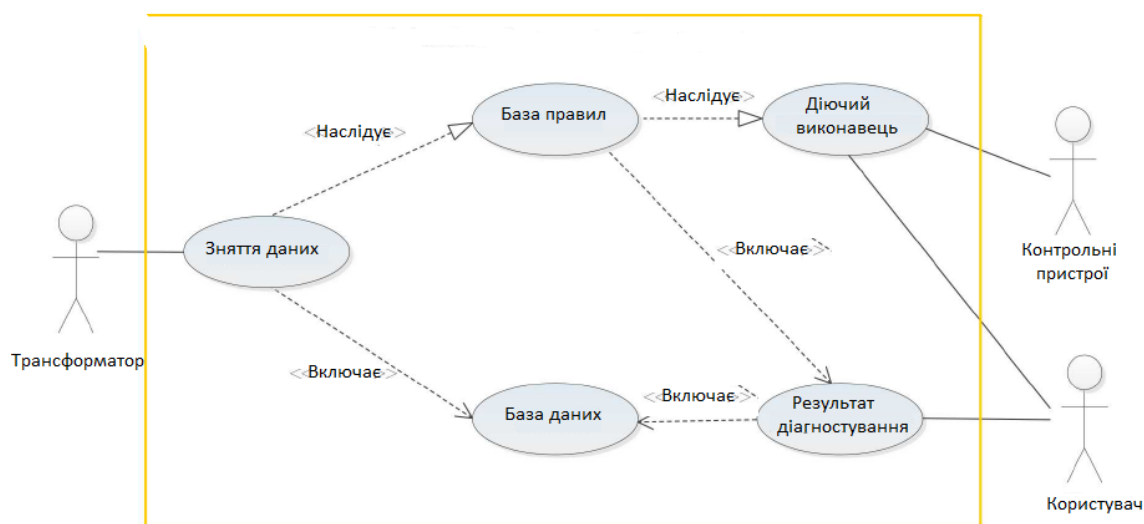


Рисунок 2.2. Діаграма діагностування трансформатора на основі правил

2.4 Діагностика дефектів трансформатора на основі онтологій

Знання в системі знань можуть бути представлені в термінах логіки. У цьому випадку набір понять всередині домену та їх логічні зв'язки між парами понять визначаються як онтологія.

Онтологія забезпечує спільне та загальне розуміння даних, що існують в межах проблемного домену інтеграції програм, та способу полегшення спілкування між людьми та інформаційними системами.

Таким чином, концепція онтології може бути використана для організації та обміну інформацією, управління знаннями та покращення взаємодії систем зв'язку в межах компанії. На основі запропонованої мультиагентної структури онтологія здатна представляти трансформатор та його компоненти [22].

Онтологія застосовується для діагностики дефектів трансформатора. У цій системі розроблена онтологія, яка відображає взаємозв'язок компонентів трансформатора, ознак несправності та типів несправностей. Мета онтології полягає в тому, щоб забезпечити обмін та повторне використання знань.

Розроблена онтологія діагностики дефектів трансформатора побудована на власній платформі і загорнута розробленим агентом для взаємодії з багатоагентною системою.

Розробка онтології діагностики дефектів трансформатора може бути вдосконалена за допомогою нечіткої онтології. Перевага нечіткої онтології полягає в тому, що її елементи можуть певною мірою належати до набору. Це допомагає певним чином визначити, чи є поняття певною категорією.

2.5 Взаємодія агентів

Спілкування агентів використовує процес управління ланцюгами організований шляхом надсилання даних у вигляді повідомлень відповідним агентам.

Відповідно до обміну повідомленнями, різні завдання можуть бути виконані за допомогою розробленої системи агента [23]. Завдання для розробленої системи перераховані наступним чином:

- Збір даних та діагностування несправностей.
- Взаємодія з користувачами.
- Виконання автоматичних дій.

Метою цього завдання є збір інформації з отриманих агентів, видалення цієї інформації та розміщення його у правильній таблиці бази даних. Статус даних також вимагає перевірки шляхом взаємодії з системою, основою на знаннях. Послідовні етапи збору даних та діагностичних завдань, а також співпраця та координація між агентами ілюструються діаграмою UML на рисунку 2.3.

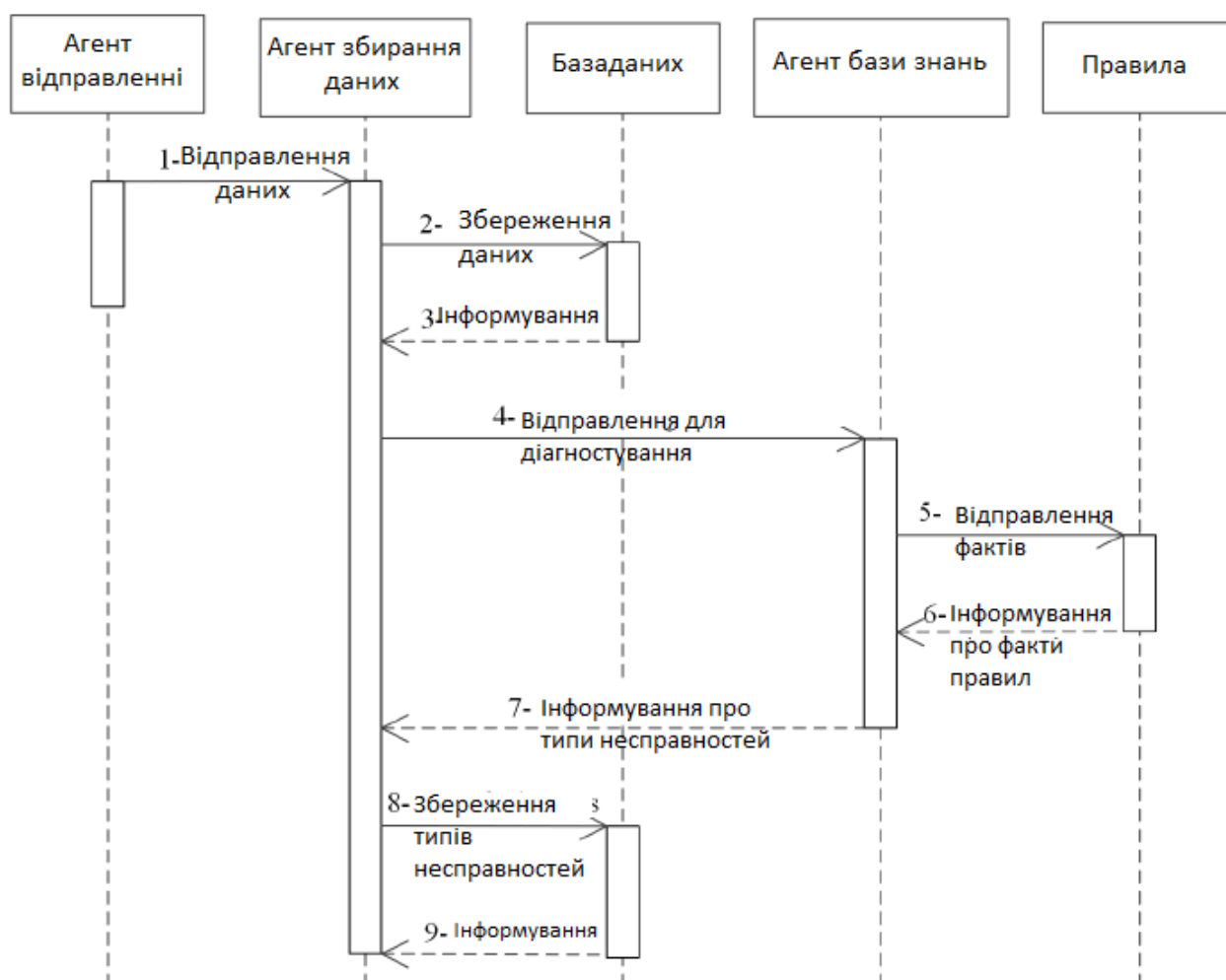


Рисунок 2.3. Взаємодія між агентами для збору даних

Метою агента збору даних є збирання отриманих даних і збереження їх у таблиці бази даних. Цей процес ініціюється агентом відправлення даних, який надсилає повідомлення, що містить дані, до агента збору даних. Агент збору даних використовує API Java для підключення до баз даних (JDBC) для підключення до бази даних. У цьому випадку база даних розроблена з MySQL і містить різні таблиці.

Дані повинні надсилатися для діагностики несправностей через агент, що базується на знаннях. Знання про конкретну інформацію написано в програмі JESS.

Дані надсилаються до JESS для виявлення несправностей, а результати (типи несправностей) відповідають на агент, який базується на знаннях. Агент, який базується на знаннях, інформує агента збору даних про типи несправностей. Нарешті, типи несправностей зберігаються в базі даних агента збору даних.

2.6 Абстрактна модель онтологій

Існують різні визначення, що даються для онтології, одна з загальноприйнятих у комп'ютерних науках: "Онтологія - це формальна, явна специфікація спільної концептуалізації".

У цьому визначенні специфікація є формальним описом того, як щось може бути побудовано для задоволення певних критеріїв, тоді як концептуалізація вимагає використання мови онтології. Це забезпечує особливу абстракцію світу та позначення цієї абстракції. Наприклад, пропозиціональна логічна формула

$$X(n) \rightarrow Y(n)$$

є абстрактною і може бути спростована і інтерпретована як:

$$\text{Mercedes} - \text{Benz}(n) \rightarrow \text{Car}(n)$$

Основні компоненти онтології підрозділяються наступним чином

- Класи
- Властивості

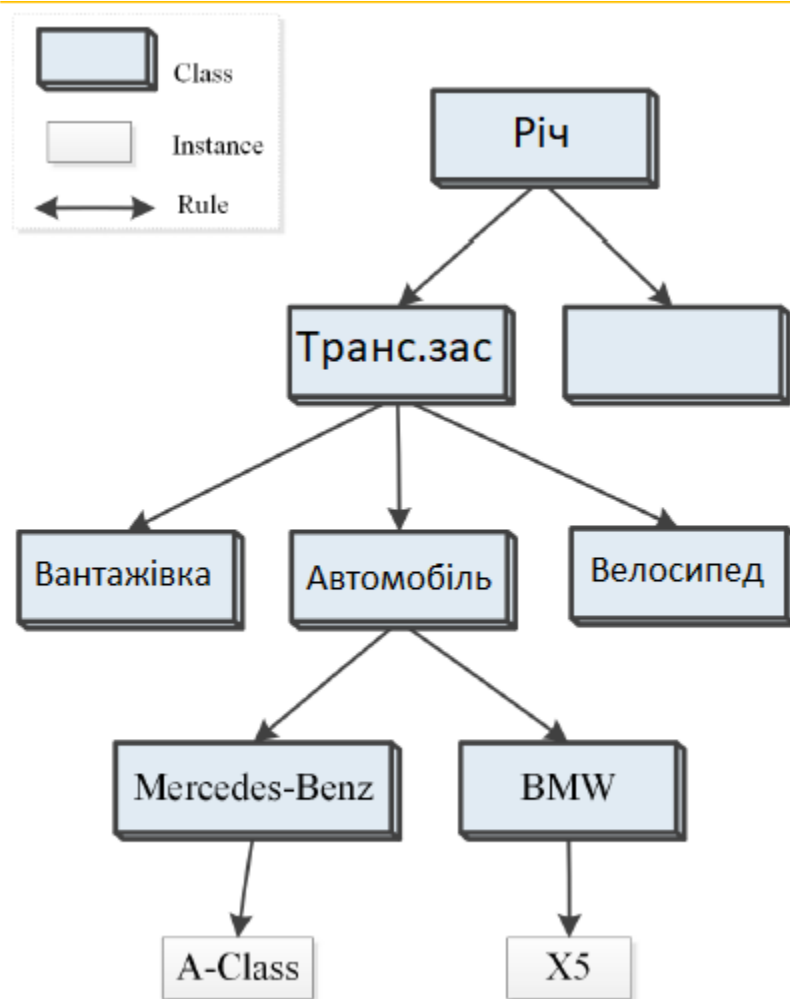


Рисунок 2.4 Графічна репрезентація онтології

Онтологія складається з класів, властивостей та окремих зразків. Класи ієрархії в онтології формуються з суперкласами та підкласами.

Різні типи властивостей забезпечують обмеження побудованої онтології. Проблеми висновку можуть бути виконані за допомогою деяких алгоритмів обґрунтування.

Алгоритм Tableau - це одне з найбільш широко використовуваних методів для аргументації в DL. Він використовується для доведення результату вирішення або обчислення складності. Розсуд дозволяє зробити висновок, заснований на побудові композиційних понять та ролей [24].

Одним із потужних підходів до побудови міркування на основі онтології є використання такого методу обґрунтування. Онтологічний розсуд виводить

запитання про базу знань онтології. Тому інтерфейс додаткового API потрібен для доступу до методології для онтології.

Оскільки трансформатор є одним з найважливіших енергосистеми, його надійність є основною проблемою в роботі енергосистеми.

Моніторинг стану трансформатору в режимі реального часу та діагностика несправностей силового трансформатора допомагають підвищити його надійність і запобігати більш серйозним проблемам.

Система діагностики несправностей з міркуваннями на основі онтології забезпечує всебічну базу знань, яку можна використати іншою програмою.

З цією метою розроблена онлайнова система діагностики несправностей, яка ґрунтується на міркуваннях онтології (рис. 2.5).

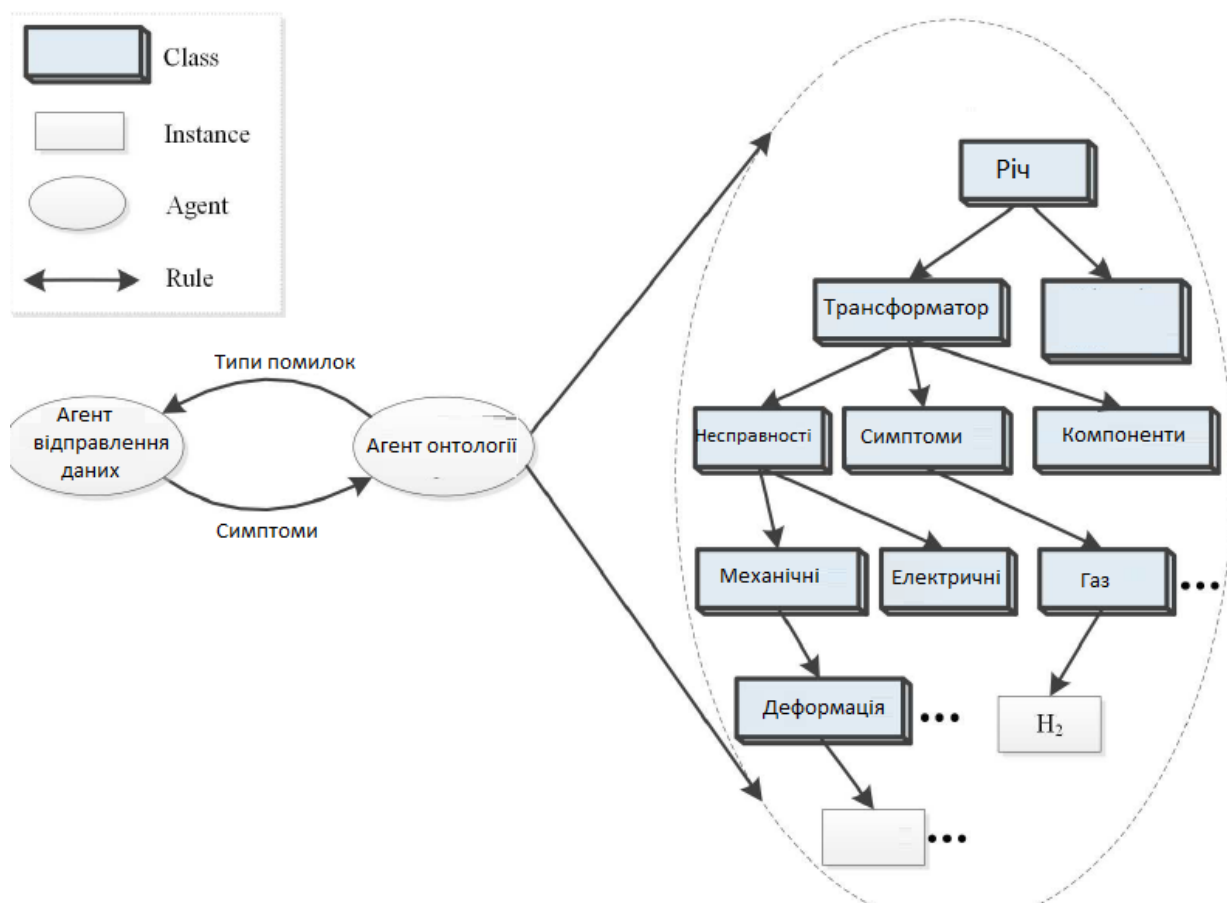


Рисунок 2.5 Мультиагентна система несправностей основана на онтології

Помилка, як правило, з'являється в силовому трансформаторі під час його роботи, однак, більш імовірно, будь-який тип несправностей може змінити робочий

стан трансформатора, що, очевидно, відображається в деяких симптомах, пов'язаних з помилкою.

Це схоже на поняття причини і наслідки, де причиною є подія, яка має відношення до його ефекту, відоме як феномен.

Іншими словами, будь-який тип помилки має деякі відповідні симптоми. Знаючи симптоми, можна виявити відповідні типи несправностей.

Наприклад, система охолодження (вентилятора) у силовому трансформаторі використовується для розсіювання тепла до зовнішнього оточення. Помилка може вплинути на робочий стан системи охолодження, що може спричинити несправність його правильної роботи (наприклад, припинення роботи вентилятора).

Це призводить до незрозумілого підвищення температури масла, представленого як симптом.

Таким чином, підвищення температури може вказувати на проблему системи охолодження, а згодом викликати зварювання.

Насправді, через складність механізму порушень трансформатора, існує безліч типів симптомів та несправностей. Для побудови відповідної онтології діагностики несправностей силового трансформатора визначаються три різні категорії, а саме: помилка, симптом і компонент.

Наприклад, помилка А має симптом В, таким чином помилка А може бути діагностована шляхом спостереження за симптомом В.

Категорія компонента також пов'язана через деякі властивості з категорією несправностей, яка відображає зв'язок між несправностями та компонентами [25]. Ці три категорії та їхні відносини використовуються як основні поняття міркування онтології для діагностики дефектів трансформатора. Елементи розробленої онтології показані на рисунку 2.6.

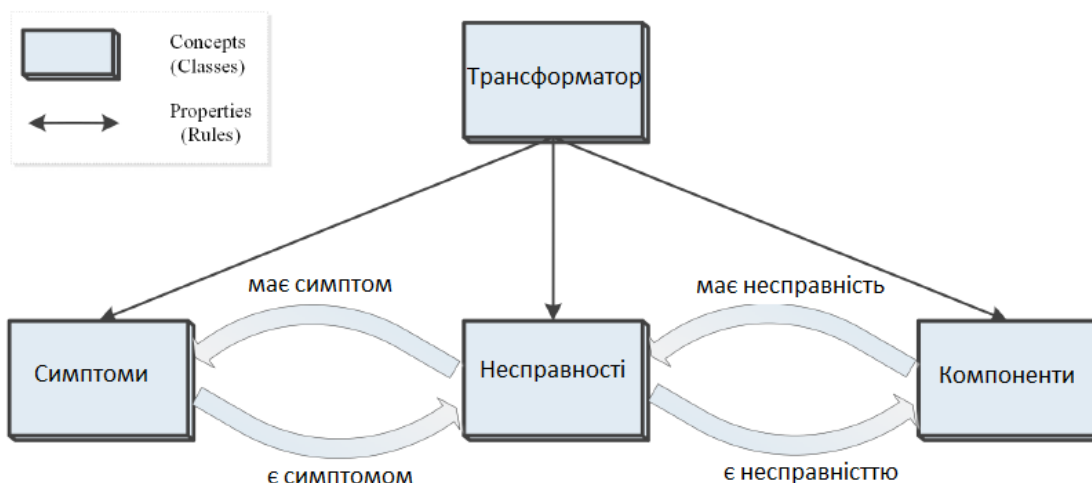


Рисунок 2.6 Головні класи онтології та їх зв'язок між собою

Виходячи з структури онтології, компоненти розробленої онтології діагностики несправностей силового трансформатора визначаються наступним чином:

Клас. Розроблена онтологія складається з трьох основних класів: компонентів, симптомів та несправностей, і, отже, кожен з них визначається як підклас класу Трансформатор.

Несправності у силовому трансформаторі можна розділити на п'ять видів: електричні, термічні, механічні, деградація та старіння. Ці п'ять типів визначаються як підкласи класу «Несправності».

Кожен тип несправностей може бути додатково поділений на різні види пов'язаних розладів.

Клас симптоми складається з різних типів симптомів, які можуть з'являтися у силовому трансформаторі. Наприклад, у методі діагностики розладу Роджера симптоми діляться на три відношення газу R1, R2 та R5, які допомагають розрізнити типи несправностей. Ці типи коефіцієнтів газу можуть бути представлені як підкласи класу Газ [26].

Можуть бути представлені інші типи симптомів, такі як кислотність, температура, електричні та фізичні симптоми, які визначаються як підкласи класу симптоми. Для випадку діагностики розладів Роджера симптоми містять підкласи GasRatios з п'ятьма типами коефіцієнтів газу R1 — R5.

Аналогічним чином, клас Компоненти містить компоненти силового трансформатора, такі як намотування, система охолодження, крани, масло тощо, визначені як його підкласи. На малюнку 2.7 показаний певний клас діагностики несправностей трансформатора.



Рисунок 2.7 Клас трансформатор та його підкласи

Розроблені класи обмежені різними типами властивостей (показані пунктирною лінією між класами на рисунку 2.7).

Властивості. Властивості забезпечують відносини класів або окремих осіб. Є два основних типи властивостей: властивості об'єкта та властивості типу даних.

Вони забезпечують різні атрибути класів. Застосовуються дві категорії властивостей, "має категорію" і "є категорія" з зворотними характеристиками один до одного.

Зворотні властивості вказують на те, що якщо властивості посиляють окремих на індивідуальний у, тоді зворотну властивість з'єднає особу у з індивідуальним х. У цьому дослідженні властивість категорії є зворотною для категорії власності.

Онтологія може бути використана для визначення суб-властивостей кожного ресурсу. У цій роботі кожне майно має три суб-властивості. Наприклад, властивість

“має категорію” має три суб-властивості: "має помилку", "має симптом" і "має компонент", з різними характеристиками [27].

Компоненти силового трансформатора можуть бути ідентифіковані відповідними типами несправностей. Наприклад, деградація ізоляції паперу в намотуванні викликає дефект, що називається зварюванням.

У разі виникнення несправності (електричний пробій) було встановлено діагноз на основі знятих симптомів, і потрібно виявити несправний компонент силового трансформатора, в якому виникла ізоляція паперу в обмотці.

2.7 Агент онтологій

Як показано на рисунку 2.5, побудована онтологія загорнута агентом, що називається агентом онтології, можна позначити як A_{ont} . A_{ont} отримує повідомлення, що містить деякі ознаки від агента, який фіксує інформацію в режимі реального часу від силового трансформатора.

У цьому випадку ми припустили, що агент даних відправника, позначений як A_{ds} , надсилає симптоми (наприклад, співвідношення газу).

A_{ont} пропускає симптоми як аргументи для онтології, і онтологія використовує свій розум, щоб перевірити несумісність класів, а також діагностувати відповідну помилку.

Як згадувалося раніше однією з основних особливостей застосування онтології є витягнення прихованої інформації з явних фактів, побудованих в онтології [28].

Для розгляду цієї ситуації подано два приклади. Несправність корозії в баку може бути діагностована на підставі високої кислотної кількості олії. Інформація, пов'язана з високим кислотним числом, може доставлятися датчиками агента, з'єднаних всередині масляного резервуара, до A_{ont} , який обгортає онтологію для діагностики несправностей.

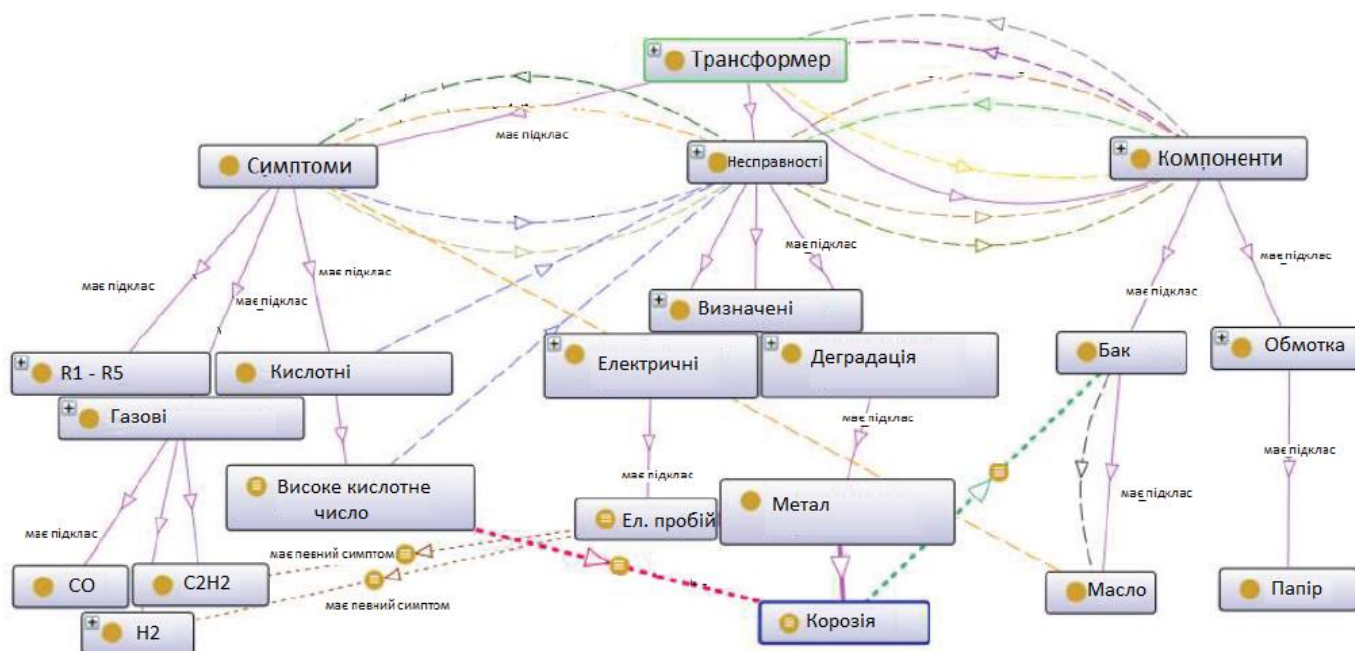


Рисунок 2.8. Онтологія для діагностування стану силового трансформатора

Запропонована онтологія, яка зображена на рисунку 2.8 має розбиття класу, що містить підклас корозія. Клас симптомів має певний підклас, високе кислотне число, і клас бак визначається як підклас компонентів класу, з підкласом масло [29].

Цікава частина методології онтології полягає в тому, що класові помилки не мають жодного підкласу під назвою "корозія баку", яка може бути виведена цією онтологією. У цьому випадку застосовані обмеження щодо несправностей, симптомів та компонентів наведені нижче: корозія має симптом тільки — висока кількість кислотного числа, що означає, що корозія є типом відмови, який можна виявити лише з симптомом високого кислотного числа. Високе кислотне число є симптомом тільки - корозії

Тобто ці дві властивості являють собою зворотнє ствердження - корозія є несправністю тільки – баку, що являє собою необхідні та достатні умови корозії з баком.

Бак має компонент -масло, що означає, що танк заповнений олією. Елементи цієї онтології показані на рисунку 2.9.

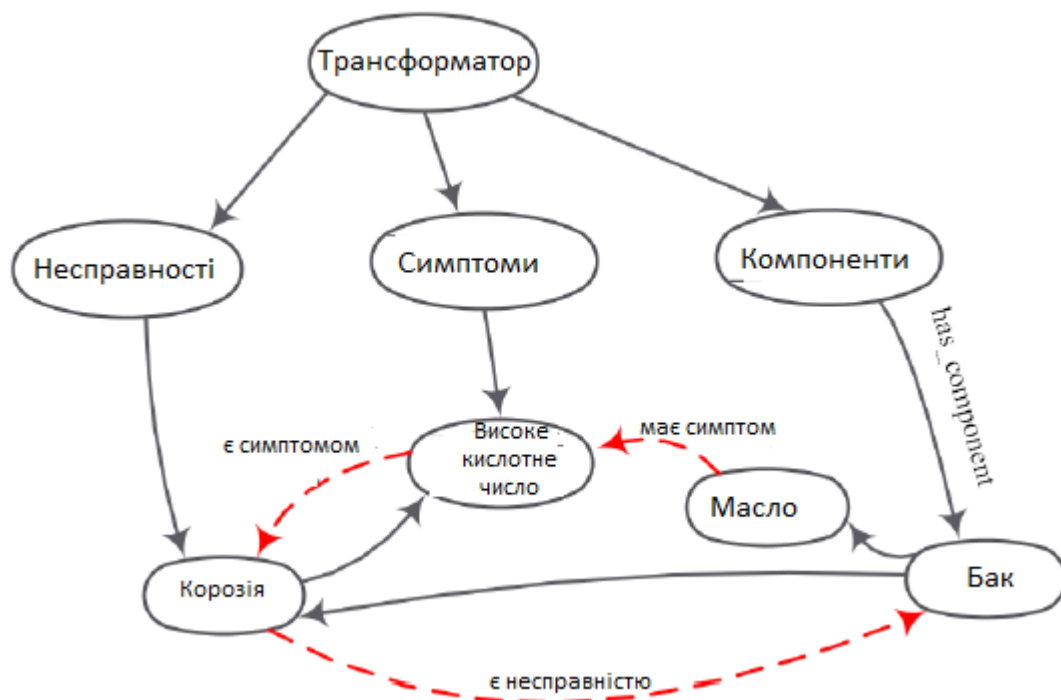


Рисунок 2.9. Приклад онтологічного методу для вилучення неявної інформації з явних фактів

Робота енергетиків включає в себе такі завдання, як оцінка стану, діагностика несправностей, обслуговування та прийняття рішень, що передбачає їхнє знання та аналіз даних.

Дана робота являє собою всеосяжну базу знань для діагностики несправностей силових трансформаторів на основі міркування онтології.

З цією метою досліджена методика Роджера з онтології для діагностики дефектів трансформатора. Резюме пропонованої онтології містить клас несправностей з вісьмома підкласами, які представляють рекомендовані випадки типів несправностей в методі Роджера [30].

Три типи властивостей типів даних застосовуються для обмеження класів несправностей. Властивості називається типовим типом даних, які містять три суб-властивості, а саме має відношення R1, має співвідношення R2 і має співвідношення R5, застосоване для цієї онтології.

2.8 Висновки до розділу

У даному розділі детальніше описано архітектуру мультиагентної системи та моніторинг стану силового трансформатора, найбільшу увагу було приділено наступним пунктам:

- Аналізу схеми моніторингу для діагностування несправностей трансформатора із відповідною структурою роботи мультиагентної системи.
- Розглянуто інтелектуальну систему, яка основана на знаннях та відповідного проведення діагностики дефектів трансформатора на основі правил, онтологій.
- Описано комунікаційні шляхи між агентами та їх взаємодію для збору та відправлення даних.
- Було продемонстровано на ілюстрації графічну репрезентацію онтології та наведено приклад онтологічного методу для вилучення неявної інформації з явних фактів.

3. ВИКОРИСТАННЯ ПРОГРАМНОГО ПРОДУКТУ ДЛЯ ІНТЕЛЕКТУАЛЬНОГО ДІАГНОСТУВАННЯ

3.1 Опис технології

Можливість бути доступною для інших систем, є ключовим значенням для системи основаної на знаннях.

Розширювана мова розмітки (XML) - це основний тип представлення знань, який забезпечує синтаксис, призначений для читання обома машинами та людьми. Її недолік полягає в тому, що представлені знання в певному домені не зрозумілі для інших програм (наприклад, програмного агента).

Семантичний веб (SW) забезпечує змістовний шар для Всесвітньої павутини (WWW), щоб машина зрозуміла. На основі представлення SW і знань, онтологія забезпечує механізм, який присвоює семантику Інтернету. Онтології також можуть використовувати різні програмні додатки.

Знання можна представити у формі логіки. В штучному інтелекті запропоновано різні види формальної логіки для представлення бази знань.

Одна з формальних дедуктивних систем, названа логікою першого замовлення (FOL), дозволяє передбачати і кількісно висунути логіку пропозицій.

Опис логіки на онтології забезпечує сильне структуроване представництво знань, яке можна зрозуміти іншими програмами. Перевірка авансових послуг, таких як семантичний пошук та автоматизоване обґрунтування, дозволяє застосовувати її різноманітно.

3.2 Засоби реалізації мультиагентної системи

3.2.1. Мова програмування: Java

Об'єктно-орієнтована мова програмування, випущена компанією Sun Microsystems у 1995 році як основний компонент платформи Java.

Синтаксис мови багато в чому походить від C та C++. Java програми виконуються у середовищі віртуальної машини Java. Java програми компілюються у байткод, який при виконанні інтерпретується віртуальною машиною для конкретної платформи.

В 2009 році Sun Microsystems придбала компанія Oracle, яка продовжує розвивати Джаву.

Розробляючи мову спеціалісти керувались п'ятьма завданнями, які були перетворені в наступні принципи.

- простота в використанні, об'єктна орієнтованість і легкість вивчення;
- надійність і безпечність;
- незалежність від архітектури;
- можливість інтерпретації;
- інтерактивність і динамічність.

Можливість JAVA виконувати свій код налюбій із підтримуваних платформ, досягається за рахунок того що її програми транслюються в проміжне уявлення, яке називається байт кодом. Байт код може інтерпретуватись влюбій системі, в якій присутнє середовище виконання JAVA.

Значна кількість ранніх систем, в якій вводилась можливість незалежності від платформи, зіштовхувалась з недостатком заключеному в зниженні продуктивності.

Байт код легко переводиться безпосередньо в рідні машинні коди, незалежно на те що в JAVA використовується інтерпретатор при якому досягається дуже висока продуктивність.

Інструкції java bytecode схожі з інструкціями машинного коду, але призначені для віртуальної машини і створені спеціально для апаратного власника.

Кінцеві користувачі приміняють як правило Java Runtime Environment (JRE конфігурація віртуальної машини, необхідна для виконання Java додатків).

Віртуальна машина Java (Java Virtual Machine) це ключовий компонент платформи Java, представляючий собою віртуальне середовище, віртуальний компютер з характеристиками нормального компютера (реєстром, кодами, набором інформації) який керує додатками Java.

Віртуальна машина Java (Java Virtual Machine) інтерпретує і виконує байт код Java, передчасно створений із початкового тексту Java програми компілятором Java.

Один із ключових принципів розробки мови Java полягав у забезпеченні захисту від несанкціонованого доступу.

Програми на Java не можуть викликати глобальні функції й одержувати доступ до довільних системних ресурсів, що забезпечує в Java рівень безпеки, недоступний для інших мов.

Даний рівень безпеки виконання Java-програм забезпечує віртуальна машина Java, котра вбудована в операційну систему. Об'єктна модель у Java проста і легко розширюється, у той же час, заради підвищення продуктивності прості типи даних Java не є об'єктами.

Java обмежує вас у декількох ключових областях і в такий спосіб сприяє виявленню помилок на ранніх стадіях розробки програми. У той же час у ній відсутні багато джерел помилок, властивих іншим мовам програмування.

Більшість використовуваних сьогодні програм “відмовляють” в одній із двох ситуацій: при виділенні пам'яті, або при виникненні виняткових ситуацій.

У традиційних середовищах програмування при розподілі пам'яті програмісту приходитьса самому стежити за усією використовуваною в програмі пам'яттю, не забуваючи звільняти її в міру того, коли вона стає зайвою.

Найчастіше програмісти забувають звільняти захоплену ними пам'ять або, що ще гірше, звільняють ту пам'ять, що усе ще використовується якою-небудь частиною програми.

Виняткові ситуації в традиційних середовищах програмування часто виникають у таких, наприклад, випадках, як ділення на нуль або спроба відкрити неіснуючий файл, і їх приходится обробляти за допомогою складних конструкцій.

Java фактично знімає обидві ці проблеми, використовуючи збирач сміття для звільнення незайнятої пам'яті й убудовані об'єктно-орієновані засоби для обробки виняткових ситуацій.

3.2.2. SpringBoot

Spring Framework (або коротко Spring) - універсальний фреймворк з відкритим вихідним кодом для Java-платформи. Також існує форк для платформи .NET Framework, названий Spring.NET.

SpringBoot є проектом на рівні IO Execution (рівень виконання) IO Spring Framework.

SpringBoot це наступний крок Spring, щоб зробити його легше в налаштуванні і розвитку додатків. З Spring Boot конфігурації Spring мінімізується максимально.

Spring Boot підтримує вбудований контейнер (embedded containers), який дозволяє веб-додаткам працювати незалежно і без необхідності застосування на Web Server.

Присутня можливість використовувати SpringBoot щоб створити Java Web додаток, що працює через команду line "Java -jar" або експортувати War файл для застосування на Web Server як зазвичай. SpringBoot дає вам "CLI Tool" для запуску сценаріїв Spring (spring scripts)

Spring Boot можна пояснити просто ілюстрацією на рисунку 3.1:

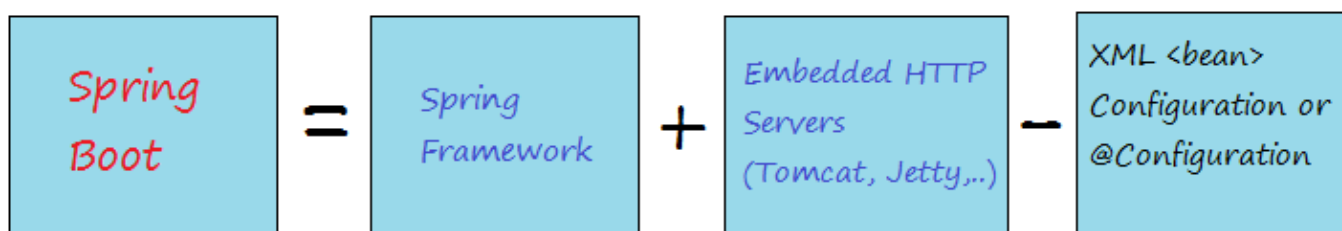


Рисунок 3.1. Архітектура SpringBoot

Переваги SpringBoot перераховані нижче:

- Легко використовується для розвитку програми на основі Spring з Java або Groovy Spring;
- Мінімізує час розвитку і піднімає продуктивність;
- Уникає написання багатьох кодів прототипу (boilerplate), Annotations і конфігурації XML;
- Легко дозволяє вам взаємодіяти з додатками Spring Boot з екологічними системами Spring як Spring JDBC, Spring ORM, Spring Data, Spring Security;
- Слід підходу "Принципи конфігурації за замовчуванням" щоб мінімізувати час і старання, вкладені для розвитку додатків.
- Забезпечує вбудований Server (Embedded HTTP servers) як Tomcat, Jetty щоб швидко і легко розвивати і тестувати веб-додатки.

3.2.3. Архітектура агентів

Термін "агент" або програмний агент знайшов свій шлях до низки технологій і широко використовувався, наприклад, в штучному інтелекті, базах даних, операційних системах та комп'ютерній мережі.

На даний момент створюються мультиагентні системи, які можуть моделювати складні системи та вводити можливості агентів, що мають спільні або суперечливі цілі.

Ці агенти можуть взаємодіяти один з одним як побічно (діючи на навколишнє середовище), так і безпосередньо (через спілкування та ведення переговорів).

Крім того, при необхідності агент може бути мобільним, з можливістю переміщення між різними вузлами в комп'ютерній мережі.

Архітектура агента є фундаментальним механізмом, що лежить в основі автономних компонентів, які підтримують ефективну поведінку в реальних, динамічних та відкритих середовищах. Справді, початкові зусилля в галузі

обчислень на основі агентів зосереджені на розробці архітектур інтелектуальних агентів, і в перші роки встановили кілька міцних стилів архітектури.

Одним з ключових компонентів мультиагентних систем є зв'язок. Фактично, агенти повинні мати можливість спілкуватися з користувачами, з системними ресурсами та один з одним, якщо їм потрібно співпрацювати, вести переговори тощо.

Багатоагентні системи можуть бути реалізовані за допомогою будь-якої мови програмування. Зокрема, об'єктно-орієнтовані мови вважаються підходящим засобом, оскільки поняття агента не дуже віддалене від поняття об'єкта.

Насправді, агенти діляться багатьма властивостями з такими об'єктами, як інкапсуляція, а часто і спадкування та передача повідомлень.

Важливою характеристикою, яку мультиагентні системи мають забезпечити, є здатність підтримувати сумісність між застарілими програмними системами. Тому наявність програмних засобів для їх інтеграції з іншими загальними технологіями може бути ключем до їх успіху.

Інтернет є одним з найважливіших областей застосування, а найважливіші засоби комунікації означають, що багатоагентні системи можуть використовуватись для забезпечення сумісності між застарілими програмними системами; отже, велика кількість поточних досліджень та розробок спрямована на забезпечення належних методів та програмних засобів для інтеграції багатоагентних систем з веб-технологіями, такими як, наприклад, Web-сервіси та Semantic Web-технології.

3.2.4. Мова програмування: JavaScript

JavaScript — кроссплатформна об'єктно-зорієнтована мова сценаріїв (скриптів). Рушій JavaScript підключається до об'єктів свого середовища виконання (зазвичай, веб-переглядача) та надає можливість керування ними.

Мова JavaScript має стандартну бібліотеку об'єктів (таких як Array, Date та Math) і основний набір елементів мови програмування, таких як оператори, керівні структури та вирази.

Ядро JavaScript може бути розширене для різних потреб шляхом доповнення його додатковими об'єктами. Наприклад:

- На стороні клієнта JavaScript розширює ядро мови, додаючи об'єкти керування переглядачем і його об'єктною моделлю документа — Document Object Model (DOM). Наприклад, клієнтські розширення дозволяють застосункам розміщувати елементи на HTML-формі та реагувати на дії користувача, такі як клацання миші, введення даних у форму і пересування сторінками.
- На стороні сервера JavaScript розширює ядро мови шляхом додавання об'єктів, що стосуються роботи JavaScript на сервері. Наприклад, серверні розширення дозволяють застосункам взаємодіяти з базою даних, забезпечувати безперервність потоку інформації від одного запущеного застосунку до іншого, або виконувати маніпуляції з файлами на сервері

JavaScript спочатку створювався для того, щоб зробити web-сторінки «живими». Програми на цій мові називаються скриптами. У браузері вони підключаються безпосередньо до HTML і, як тільки завантажується сторінка - тут же виконуються.

Програми на JavaScript - звичайний текст. Вони не вимагають якоїсь спеціальної підготовки.

В цьому плані JavaScript сильно відрізняється від іншої мови, наприклад Java.

Сучасний JavaScript - це «безпечна» мова програмування загального призначення. Він не надає низькорівневих засобів роботи з пам'яттю, процесором, так як спочатку був орієнтований на браузери, в яких це не потрібно.

Мови програмування JavaScript та Java у окремих рисах схожі між собою, але все одно докорінно відрізняються одна від одної. Мова JavaScript чимось нагадує Java, але, на відміну від останньої, не має чіткої типізації даних та строгого контролю за типами.

JavaScript наслідує синтаксис більшості виразів Java, також наслідуються угоди про іменування та основні конструкції керування перебігом виконання сценарію. Це і стало причиною того, що свого часу назву мови змінили з LiveScript на JavaScript.

На відміну від системи часу коміляції декларативно побудованих класів у Java, мова JavaScript підтримує систему часу виконання, що ґрунтується на невеликій кількості типів даних, представлених числовим типом, логічними, та рядковими значеннями.

JavaScript має об'єктну модель, що спирається на прототипи, замість більш поширеної, побудованої на класах, об'єктної моделі, властивій мові Java. Модель на базі прототипів надає можливість динамічного спадкування; тобто успадковане може бути різним для різних об'єктів.

Також у JavaScript реалізована підтримка функцій без будь-яких особливих декларативних вимог. Функції можуть бути властивостями об'єктів, функціями, що виконуються, та слабко типізованими методами.

Для JavaScript властива вільніша форма мови, ніж для Java. Вас не зобов'язують оголошувати всі змінні, класи та методи. JavaScript-програміста не турбує питання, є метод відкритим (public), закритим (private), чи захищеним (protected). Для змінних, параметрів, та типів, що повертаються функціями, не вказується явним чином їх тип.

У даній роботі на мові Java було реалізовано backend (логічну частину проекту) та на мові JavaScript відображення графіків та відображення елементів на сторінці.

3.3 Принцип роботи

Уповноважений працівник підприємства у якого присутній доступ до даної системи інтелектуального діагностування має змогу відкрити даний програмний продукт у браузері.

На головному екрані буде знаходитись перелік можливих функцій для користувача, як зображено на рисунку 3.2.



Рисунок 3.2. Головна сторінка програмного продукту

Наступним кроком для інтелектуального діагностування стану силового трансформатора користувачу надається вибір різних методів діагностування:

- хроматографічний аналіз розчинених у маслі газів;
- хроматографічний аналіз розчинених у маслі газів герметичних вводів;
- прилади реагування напруги під навантаженням;
- контроль високовольтних вводів;
- фізико-хімічний аналіз масла із баку трансформатору;
- технічний стан силового трансформатору.

Дану можливість відображено на рисунку 3.3.

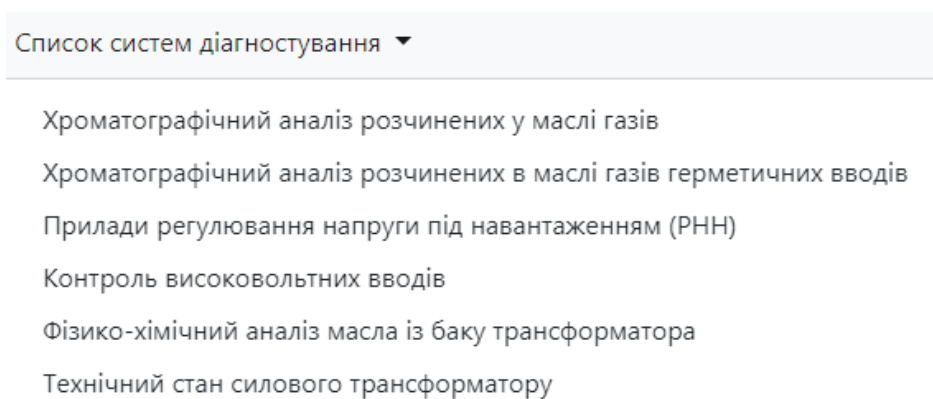


Рисунок 3.3. Методи діагностування силового трансформатору

Після вибору одного із методів для відповідного діагностування користувач буде направлений на відповідно сторінку, на якій матиме змогу як ввести власноруч зняті показники із компонентів силового трансформатору так і одразу провести опрацювання даних, які були завантажені із онлайн моніторингу (рис. 3.4).

Хроматографічний аналіз розчинених у маслі газів

Концентрація газів водню H ₂	0,723
Концентрація газів метану CH ₄	1,987
Концентрація газів ацетилену C ₂ H ₂	0,87
Концентрація газів етану C ₂ H ₆	7,174
Концентрація газів етилену C ₂ H ₄	2,976
Концентрація газів оксиду вуглецю CO	0,521
Концентрація газів діоксиду вуглецю CO ₂	1,98
Відносна швидкість зростання газу	0,33

[Відправити дані на опрацювання](#)
[Візуалізація результатів діагностування](#)

Рисунок 3.4. Система для вводу даних із онлайн моніторингу

Після відповідно введених даних або тих, що були завантажені попередньо, користувач повинен відправити ці дані на опрацювання.

Внаслідок опрацювання цих даних результат буде відображений у вигляді переліку несправностей, які були виявлені в даному трансформаторі, симптомів які призвели до даних несправностей, та які можуть призвести до майбутніх несправностей, а також список аварійних компонентів, які варто негайно замінити для правильного функціонування трансформатору (рис. 3.5).

Результати діагностування

<p>Список помилок (несправностей)</p> <p>Помилка концентрації етану у маслі, що викликає перегрів до 300 градусів.</p> <p>Помилка концентрації діоксиду вуглецю у маслі, що викликає деградацію заліза.</p> <p>Помилка концентрації ацетилену у маслі.</p> <p>Помилка концентрації метану у маслі.</p> <p>Помилка концентрації етилену у маслі, що викликає осьові деформації.</p> <p>Помилка концентрації водню у маслі, що викликає перегрів до 200 градусів</p> <p>Помилка концентрації оксиду вуглецю у маслі, що викликає перегрів до 150 градусів.</p> <p>Список симптомів</p> <p>Помилка із перевищенням допустимого вмісту етану виникла через збільшення його вмісту, допустима норма 0.005% відносно об'єму масла.</p> <p>Помилка із перевищенням допустимого вмісту водню виникла через збільшення його вмісту, допустима норма 0.01% відносно об'єму масла.</p> <p>Помилка із перевищенням допустимого вмісту діоксиду вуглецю виникла через збільшення його вмісту, допустима норма 0.8% відносно об'єму масла.</p> <p>Помилка із перевищенням допустимого вмісту етилену виникла через збільшення його вмісту, допустима норма 0.001% відносно об'єму масла.</p> <p>Помилка із перевищенням допустимого вмісту оксиду вуглецю виникла через збільшення його вмісту, допустима норма 0.06% відносно об'єму масла.</p> <p>Помилка із перевищенням допустимого вмісту метану виникла через збільшення його вмісту, допустима норма 0.01% відносно об'єму масла.</p> <p>Помилка із перевищенням допустимого вмісту ацетилену виникла через збільшення його вмісту, допустима норма 0.001% відносно об'єму масла.</p> <p>Список пошкоджених компонентів</p> <p>Пошкоджений бак з маслом</p> <p>Пошкоджена електроізоляційна бумага</p> <p>Пошкоджений рівень рідини</p> <p>Пошкоджений сердечник трансформатора</p> <p>Пошкоджені обмотки</p>

Рисунок 3.5. Результати інтелектуального діагностування

Відповідно до проведеного інтелектуального діагностування користувач має можливість відслідкувати глибину ураження силового трансформатору, нажавши на кнопку “Візуалізація результатів діагностування”, яка зображена на рисунку 3.4.

В результаті чого відображається відповідний графік (рис. 3.6).



Рисунок 3.6. Графічна візуалізація результатів діагностування

3.4 Висновки

У даному розділі описано використання програмного продукту для інтелектуального діагностування стану силового трансформатора.

На початку роботи було здійснено короткий опис використовуваних технологій.

Після чого було проаналізовано всі можливі варіанти щодо вибору відповідної мови програмування для реалізації даного продукту, тож було вибрано наступний стек технологій: мови програмування Java та JavaScript, SpringBoot фреймворк з відкритим вихідним кодом для Java-платформи.

Було продемонстровано архітектуру програмного агента, який буде використовуватись для створення мультиагентної системи.

Одним з ключових компонентів мультиагентних систем є зв'язок. Фактично, агенти повинні мати можливість спілкуватися з користувачами, з системними

ресурсами та один з одним, якщо їм потрібно співпрацювати, вести переговори тощо.

В кінці даного розділу було продемонстровано принцип роботи даного програмного продукту з відповідними ілюстраціями для більш сприйнятливого розуміння.

4. РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ “ІНТЕЛЕКТУАЛЬНЕ ДІАГНОСТУВАННЯ СИЛОВОГО ТРАНСФОРМАТОРУ”

4.1 Опис ідеї проекту

У даному розділі описано економічне обґрунтування реалізації стартап-проекту на тему “Інтелектуальне діагностування силового трансформатору”.

Система діагностування буде реалізована у вигляді програмного забезпечення, тому у користувачів буде можливість використовувати його на підприємстві при доступі до інтернету.

Процес реалізації включає в себе:

- імплементацію архітектурного рішення попередньо згаданого у роботі;
- створення відповідно перспективної стратегії для виходу конкурентноспроможного програмного забезпечення на ринок.

Опис головної ідеї стартап-проекту наведено в Таблиці 4.1

Таблиця 4.1 — Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Ідея полягає в тому, щоб створити програмне забезпечення для інтелектуального діагностування силового трансформатору на основі онтологій	1. Підвищення швидкості реагування на несправності трансформатора	Працівники швидше відреагують та усунуть пошкоджений компонент
	2. Полегшення роботи працівників	Працівнику не потрібно ходити та власноруч знімати показання з усіх компонентів трансформатору

В табл. 4.1 було показано основну ідею, напрямки застосування та вигоди для користувача стартап-проекту. Дане програмне забезпечення отримує дані по яких відбуватиметься інтелектуальне діагностування із онлайн моніторингу силових трансформаторів.

В таблиці 4.2 наведено відповідне порівняння із потенційними конкурентами, дана таблиця надає можливість оцінити конкурентоспроможність та можливості виходу стартапу на ринок.

Таблиця 4.2 — Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко- економічні характеристики ідеї	(Потенційні) товари/концепції конкурентів				W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	Конкурент1	Конкурент2	Конкурент3			
1.	Форма виконання	Веб-сервіс	Сервіс	Сервіс	Сервіс			+
2.	Собівартість	низька	Висока	Висока	Висока			+
3.	Наявність адміністратора для налаштування	Не треба	Так	Так	Так			+
4.	Наявність інтернету	Треба	Ні	Ні	Ні			+
5.	Кросплатформеність	Так	Ні	Ні	Ні		+	

Визначений перелік слабких, сильних та нейтральних характеристик та властивостей ідеї потенційного товару є підґрунтям для формування його конкурентоспроможності.

Сильними сторонами є форма виконання, собівартість, наявність адміністратора для налаштування та наявність інтернету, а нейтральною – крос-платформеність.

4.2 Технологічний аудит ідеї проекту

В межах даного підрозділу необхідно провести аудит технології, за допомогою якої можна реалізувати ідею проекту (технології створення товару).

Таблиця 4.3 — Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технології	Доступність технології
1.	Створення веб-сервісу	Java, SpringBoot	Наявна	Безкоштовна, доступна
Обрана технологія реалізації ідеї проекту: для створення веб-сервісу обрані технології Java, SpringBoot, які є безкоштовними, доступними та добре дослідженими потенційними розробниками				

Отже, даний проекту буде реалізована на основі технології Java та додаткового використання SpringBoot – ці два стека технологій будуть використані з метою реалізації веб-сервісу для інтелектуального діагностування.

4.3 Аналіз ринкових можливостей

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Спочатку проводимо аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку.

Таблиця 4.4 - Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1.	Кількість головних гравців, од	3
2.	Загальний обсяг продаж, грн/ум.од	10000 грн./ум.од
3.	Динаміка ринку (якісна оцінка)	Зростає
4.	Наявність обмежень для входу (вказати характер обмежень)	Немає
5.	Специфічні вимоги до стандартизації та сертифікації	Конфіденційність оброблюваних даних
6.	Середня норма рентабельності в галузі (або по ринку), %	$R = (3000000 * 100) / (1000000 * 12) = 25\%$

Після проведеного аналізу варто стверджувати, що обмеження для входу на ринок відсутні, динаміка ринку зростає, галузь є рентабельною. Проте варто врахувати дії, які необхідні виконати для потенційних споживачів даного продукту, які наведено у Таблиці 4.5.

Таблиця 4.5 - Характеристика потенційних клієнтів стартап-проекту

№	Потреба, що	Цільова аудиторія	Відмінності у	Вимоги
---	-------------	-------------------	---------------	--------

п/ п	формує ринок	(цільові сегменти ринку)	поведінці різних потенційних цільових груп клієнтів	споживачів до товару
1.	Веб-сервіс для інтелектуального діагностування трансформаторів	Будь-які підприємства у яких в обладнанні присутній силовий трансформатор	Цільова група працівників на підприємстві, яка займається діагностуванням трансформаторів	Надійність та конфіденційність користуванням

Відповідно до проведеної характеристики цільових клієнтів даного продукту, наведеного у Таблиці 4.5 та необхідністю виходу продукту на ринок, який було описано у Таблиці 4.4, варто спрямувати всі сили на активне просування проекту в усіх підприємства де використовуються силові трансформатори.

Наступним кроком потрібно скласти таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (табл. 4.6 — 4.7).

Таблиця 4.6 - Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Динаміка ринку	Уповільнення росту ринку	Розширення на суміжні ринки

2.	Конкуренція	Вихід на ринок великої компанії	Надати додаткові переваги власного продукту лише за появи сильного конкурента
3.	Потреби користувачів	Користувачам необхідний сервіс з глибшим функціоналом	Надання нового функціоналу вже існуючій системі (завдяки її адаптивності)
4.	Збільшення витрат на технічну підтримку	Невчасне реагування на сучасний ринок потреб користувачів	Вчасно оновлювати програмне забезпечення

Отже, було проаналізовано фактори загроз ринкового впровадження проекту, серед яких: динаміка ринку, конкуренція, потреби користувачів, збільшення витрат на технічну підтримку звідки видно ряд ризиків, які слід врахувати при плануванні виходу на ринок та мати орієнтовні сценарії їх мінімізації та компенсування їх впливу.

Аналогічно до загроз стартап-проекту, що можуть спричинити значні проблеми для його розвитку, важливою частиною є огляд можливих сприятливих умов, використання яких може значно покращити становище спартап-проекту та надати перевагу порівняно із конкурентами. Такі сприятливі умови та відповідні можливості розглянуто в Таблиці 4.7.

Таблиця 4.7 - Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1.	Зростання можливостей потенційних покупців	Зростання фінансування у підприємств, які	Запропонувати їм свої послуги

		використовують силові трансформатори	
2.	Зниження довіри до конкурента 1	У додатку конкурента 1 нещодавно було знайдено витік інформації, яка збиралася для аналітики	При виході на ринок звертати увагу покупців на безпеку нашого додатку
3.	Зростання попиту користувачів на аналогічне програмне забезпечення	Збільшення кількості користувачів в сфері Інтернету речей	Запропонувати користувачам свій програмний продукт

У таблиці 4.7 наведено фактори можливостей ринкового впровадження проекту, серед яких: зростання можливостей потенційних покупців, зниження довіри до конкурента та зростання попиту користувачів на аналогічне програмне забезпечення.

Надалі проводиться аналіз пропозиції: визначаються загальні риси конкуренції на ринку.

Таблиця 4.8 - Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції: - досконале	Існує 2 фірми конкуренти на ринку	Врахувати ціни конкурентних компаній на початкових етапах створення бізнесу, реклама (вказати на конкретні переваги перед

		конкурентами)
2. За рівнем конкурентної боротьби: - міжнародний	Один зарубіжний конкурент	Додати можливість вибору мови ПЗ, щоб легше було у майбутньому вийти на міжнародний ринок
3. За галузевою ознакою: - внутрішньогалузева	Конкуренти мають ПЗ, який використовується лише всередині даної галузі	Створити основу ПЗ таким чином, щоб можна було легко його переробити для використання у інших галузях
4. Конкуренція за видами товарів: - товарно-видова	Види товарів є однаковими, а саме - програмне забезпечення	Створити ПЗ, враховуючи недоліки конкурентів

Продовження таблиці 4.8.

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
5. За характером конкурентних переваг: - нецінова	Вдосконалення технології створення ПЗ, для зменшення його собівартості	Використання менш дорогих технологій, більш ефективних методологій
6. За інтенсивністю: - не марочна	Бренди відсутні	-

В даній таблиці (табл. 4.8) було продемонстровано ступеневий аналіз конкуренції на ринку, де було визначено особливості конкурентного середовища та їх вплив а діяльність підприємства.

Після аналізу конкуренції проводиться більш детальний аналіз умов конкуренції в галузі (табл. 4. 9)

Таблиця 4.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Навести перелік прямих конкурентів	Визначити бар'єри входження в ринок	Визначити фактори сили постачальників	Визначити фактори сили споживачів	Фактори загроз з боку замінників
Висновки	Існує 2 конкуренти на ринку. Найбільш схожим за виконанням є конкурент 1.	Так, можливості для входу на ринок є, бо наше рішення спрощує та пришвидшує роботу спеціаліста.	Постачальники відсутні	Важливим для користувача є зручність у користуванні	Товари-замінники можуть використати більш дешеву технологію створення додатку та зменшити собівартість товару

Виходячи з аналізу конкурентоспроможності можна впевнено стверджувати, що проект має можливість роботи на ринку, тому що серед наведених конкурентів

немає тих, які б могли його потіснити, адже розроблене рішення спрощує та пришвидшує роботу спеціаліста.

На основі аналізу конкуренції із урахуванням характеристик ідеї проекту, вимог споживачів до товару та факторів маркетингового середовища (Таблиці 4.8 та 4.9) визначається та обґрунтовується перелік факторів конкурентоспроможності що наведено у Таблиці 4.10

Таблиця 4.10 - Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1.	Використання ПЗ у вигляді веб-сервісу	Дозволяє наочно побачити роботу ПЗ.
2.	Простота інтерфейсу користувача	Перехід по відповідним вкладкам у сервісі

Було наведено основні фактори конкурентоспроможності, які будуть представлені на ринку, а саме: використання технології пов'язаної з мультиагентною системою для отримання даних із силового трансформатору, простота користувацького інтерфейсу.

За визначеними факторами конкурентоспроможності проводиться аналіз сильних та слабких сторін стартап-проекту (табл. 4.11).

Таблиця 4.11 - Порівняльний аналіз сильних та слабких сторін проекту

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з нашим підприємством						
			-3	-2	-1	0	+1	+2	+3

1.	Використання ПЗ у вигляді веб-сервісу	17		+					
2.	Простота інтерфейсу користувача	20	+						
3.	Необхідність в доступі до Інтернету	15						+	

В табл. 4.11 було здійснено порівняльний аналіз сильних сторін проекту товарів-конкурентів і нашого підприємства. Найбільше балів набрано для таких факторів конкурентноспроможностей - простота інтерфейсу користувача, використання ПЗ у вигляді веб-сервісу.

Далі складаємо SWOT-аналіз (табл. 4.12) . Результат SWOT-аналізу представлено у вигляді матриці аналізу сильних (Strength), слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) стартап-проекту

Таблиця 4.12 – SWOT-аналіз стартап-проекту

S	Можливість надання нового функціоналу вже існуючій системі, відсутність жорстких вимог до платформи	Необхідність в доступі до Інтернету	W
O	Маркетингова реклама, акцентування уваги на надійності системи, заохочення співробітників конкурентів до зміни компанії	Надання додаткових переваг власного продукту лише за появи сильного конкурента, перегляд виконання умов, що зменшують податки, поступове підвищення тарифів	T

На основі SWOT-аналізу розроблено альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок.

Для визначених альтернатив виконано аналіз з точки зору строків та ймовірності отримання ресурсів, результати якого наведені в Таблиці 4.13

Таблиця 4.13 – Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1.	Створення системи діагностування на основі нечітких онтологій	40%	8 місяців
2.	Створення системи діагностування на основі правил	30%	12 місяців

З означених альтернатив обирається та, для якої: а) отримання ресурсів є більш простим та ймовірним; б) строки реалізації – більш стислими

4.4 Розробка ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (табл. 4.14).

Таблиця 4.14 - Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1.	Підприємства	Можливість автоматичного діагностування, що зменшує роботу людей	Великий	Існує 1 конкурент, який надає схоже, але менш швидке рішення.	Швидкодія, зручний користувацьк ий інтерфейс, точність діагностики
Обрано цільові групи: Підприємства					

Проаналізувавши потенційні групи споживачів, які наведені в табл. 4.14 видно що ним являються підприємства у використанні яких залучені силові трансформатори.

Далі визначається базова стратегія розвитку (табл. 4.15).

Таблиця 4.15 – Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1.	Створення програмного продукту із використанням	Ринкове позиціонування	Швидкодія, простота у користуванні, безпечність, надійність	Диференціа ція

	мультіагентних систем			
--	-----------------------	--	--	--

Отже, була обрана альтернатива розвитку проекту - створення програмного продукту з використанням мультіагентних систем. Визначена базова стратегія розвитку - диференціація, інструментом реалізації якої є ринкове позиціонування, адже даний програмний продукт матиме основні відмінності від товарів-конкурентів такі як швидкодія, простота у користуванні, безпечність, надійність

Наступним кроком є вибір стратегії конкурентної поведінки (табл. 4.16).

Таблиця 4.16 - Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохід цем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1.	Так	Так	Ні	Зайняття конкурентної ніші

Отже, було визначено базову стратегію конкурентної поведінки - зайняття конкурентної ніші, адже програмний продукт буде націлений на один ринковий сегмент — отримання даних через онлайн моніторинг.

Також матиме технічну підтримку у вигляді оновлень даного продукту - основну перевагу перед конкурентами, що формуватиме довіру і прихильність споживачів

Далі визначається стратегія позиціонування проекту, яка допоможе користувачам ідентифікувати програмний продукт (табл. 4.17).

Таблиця 4.17 - Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап- проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1.	Простота інтерфейсу, швидкодія, надійність, безпечність	Диференціа ція	Простота користувацького інтерфейсу дозволить отримувати необхідні дані і відслідковувати події в режимі реального часу, а також буде безпечним і надійним	Швидкодія, безпека, простота

Отже, було вибрано такі асоціації, які мають сформувати комплексну позицію власного проекту

4.5 Розробка маркетингової програми

Першим кроком є формування маркетингової концепції товару, який отримає споживач. Для цього у Таблиці 4.18 потрібно підсумувати результати попереднього аналізу конкурентоспроможності товару.

Таблиця 4.18 - Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1.	Швидкодія	ПЗ працює досить швидко у зв'язку із отриманням даних через онлайн моніторинг	Перевага у швидкості
2.	Простота користувацького інтерфейсу	Простота роботи веб-сервісу	Користувачі мають зручний інтерфейс для взаємодії з програмним продуктом

Отже бачимо, що проект має ключові переваги перед конкурентами, які повністю відповідають потребам цільової аудиторії.

Далі у Таблиці 4.19 проілюстрована трирівнева маркетингова модель товару: уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання.

Таблиця 4.19 - Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові
--------------	----------------------

I. Товар за задумом	Система, яка проводить інтелектуальне діагностування стану силового трансформатора із застосування мультиагентної системи.		
II. Товар у реальному виконанні	Властивості / характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Зручність та простота користувацького інтерфейсу	Нм	Технологічна
	2. Швидкість роботи	Нм	Технологічна

Продовження таблиці 4.19

Рівні товару	Сутність та складові		
	3. Безпека відповідно до стандартів	Нм	Технологічна
	Якість: згідно до стандарту ISO 27001.		
	Маркування відсутнє		
	Моя компанія: “I.Diagnosis”		
ІІІ. Товар із підкріпленням	Тижнева пробна безкоштовна версія		
	Постійна підтримка для користувачів		
За рахунок чого потенційний товар буде захищено від копіювання: патент			

Обґрунтування кожного із рівнів наведено нижче:

1-й рівень. Задум товару - засобом вирішення якої потреби / проблеми буде даний товар, яка його основна вигода (основа технічного завдання).

2-й рівень. Рішення реальної реалізації товару: якість, властивості, упаковка.

3-й рівень. Товар з підкріпленням (супроводом) - додаткові послуги та переваги для споживача, що створюються на основі товару за задумом і товару в реальному виконанні (гарантії якості, доставка, умови оплати та ін).

Наступним кроком є визначення цінових меж (табл. 4.20). Аналіз проводиться експертним методом.

Таблиця 4.20 - Визначення меж встановлення ціни

№ п/п	Рівень цін на товари- замінники, грн	Рівень цін на товари-аналоги, грн	Рівень доходів цільової групи споживачів, грн	Верхня та нижня межі встановлення ціни на товар/послугу, грн
1.	38000	42000	320000	330000-470000

Наступним кроком є визначення оптимальної системи збуту, в межах якого приймається рішення (табл. 4.21).

Таблиця 4.21 - Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1.	Купують підписку та роблять щорічне подовження ліцензії	Продаж	0(напрям), 1(через одного посередника)	Власна та через посередників

Отже, було сформовано систему збуту у вигляді щорічної підписки (ліцензії). Збут буде проводитися власними силами та через посередників, напряму та через одного посередника у вигляді продажу товару.

Далі розробляється концепція маркетингових комунікацій (табл. 4.22).

Таблиця 4.22 - Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1.	Використання за допомогою сайту	Інтернет	Швидкодія, простота у використанні, безпека	Показати переваги сервісу, у тому числі і перед конкурентами	Демо-ролик із використання

За результатами розробки концепції маркетингових комунікацій, представленими в Таблиці 4.22, та використовуючи результати формування системи збуту, наведених в Таблиці 4.21, було визначено специфіку поведінки цільових клієнтів; Інтернет, як головний канал комунікації цільових клієнтів.

4.6 Висновки

У даному розділі були досліджені основні аспекти виходу на ринок програмного продукту для інтелектуального діагностування стану силового трансформатору із використанням мультиагентних систем.

Описаний продукт є доцільним для працівників, які хочуть мати швидке та достовірне діагностування силових трансформаторів на основі даних отриманих із онлайн моніторингу

В рамках розділу було визначено перелік сильних, нейтральних та слабких характеристик, властивостей потенційного товару для формування його конкурентоспроможності; обрана технологія реалізації ідеї проекту: для створення програмного продукту обрана технологія SpringBoot із застосуванням мультиагентних систем.

Проведений ступеневий аналіз конкуренції на ринку, SWOT аналіз та обґрунтовані фактори конкурентоспроможності; проведений менеджмент потенційних ризиків .

Отже відповідно до проведених досліджень існує можливість ринкової комерціалізації проекту. Також існують перспективи впровадження з огляду на потенційні групи клієнтів, бар'єри входження не є високими, проект має значні переваги перед конкурентами: швидкодія, простота у використанні, безпека, надійність.

Для успішного виконання проекту необхідно реалізувати програмний продукт із використанням технології мультиагентних систем. Проаналізувавши отримані результати, можна зробити висновок, що подальша імплементація є доцільною.

ВИСНОВКИ

Внаслідок виконання дипломної роботи магістра здійснювалось розробка програмного забезпечення для інтелектуального діагностування стану силового трансформатору із застосуванням мультиагентного підходу. Результати даної роботи було реалізовано в програмному продукті.

В роботі була описана загальна ієрархія розробленої багатоагентної системи для моніторингу стану силових трансформаторів та діагностики несправностей.

Розроблено різні типи агентів для моніторингу стану, контролю та виконання автоматичних дій. Вибрана система агента вирішує проблему онлайн-моніторингу та діагностики несправностей.

Було представлено МАС з міркуваннями на основі правил та онтологій для виконання дій та діагностики несправностей.

Діагностування несправностей із застосуванням бази знань на основі онтологій для силового трансформатора було проведено із застосуванням моделювання тестових даних, отриманих із онлайн моніторингу стану силового трансформатору.

Для представлення результатів інтелектуального діагностування стану силового трансформатору було розроблено функцію візуалізації даних у вигляді трикутника оцінювання стану трансформатору на основі його компонентів.

Якість діагностування в перспективі можна покращити із застосуванням бази знань на основі нечіткої онтології.

Створення МАС на базі нечіткої онтології у силовому трансформаторі - нечітка онтологія забезпечує семантичні анотації на основі логіки для вирішення невизначених знань. У дисертації описується обмежена кількість розроблених агентів.

Під час виконання даної магістерської дисертації було здійснено моніторинг та розбір даних попередньо вибраного трансформатору.

Під час виконання було розглянуто та проаналізовано декілька алгоритмів пошуку помилок із використанням мультиагентного підходу на базі знань, а саме онтологій.

Здійснено реалізацію пошуку помилок з відповідними даними взятими попередньо із виконаного моніторингу, а також можливо в цей же час здійснювати зміни у даних, які будуть надіслані на опрацювання та збережені для подальших дій.

Останнім кроком у реалізації даного програмного продукту ставилось за мету створити зручний інтерфейс для роботи із продуктом, що було успішно здійснено.

Використання розроблених засобів з базою знань у поєднанні з особливостями онтологій, описаними вище, дозволяє забезпечити додаткові можливості для використання даного програмного продукту в енергетичній системі.

ПЕРЕЛІК ПОСИЛАНЬ

1. Ковальчук А.М., Поліщук О.В. Інтелектуальне діагностування стану силового трансформатора із застосуванням мультиагентного підходу. зб. наук. праць “ΛΟΓΟΣ. The art of scientific mind” з матеріалами міжнар. наук.-практ. конф., м. Івано-Франківськ, 5 грудня, 2018р. Вінниця : ГО “Європейська наукова платформа”, 2018 Т.4. С. 77-79.
2. D. P. Buse and Q. H. Wu, IP Network-Based Multi-Agent Systems for Industria Automation. Springer, 2006.
3. Z. Yang, C. Ma, J. Q. Feng, Q. H. Wu, S. Mann, and J. Fitch, “A multi-agent framework for power system automation,” International Journal of Innovations in Energy Systems and Power, vol. 1, 2006.
4. A. Purwadi, N. Heryana, D. Nurafiat, C. Sosetyo, A. Setiana, and A. Mustaqim, “Testing and diagnostics of power transformer in pt. Indonesia power kamojang geothermal power plant unit 1,” in Electrical Engineering and Informatics (ICEEI), 2011 International Conference, pp. 1–5, 2011.
5. D. Wang, W. H. Tang, and Q. H. Wu, “Ontology-based fault diagnosis for power transformers,” in Power and Energy Society General Meeting, 2010 IEEE, pp. 1–8, 2010.
6. S. D. J. McArthur, S. M. Strachan, and G. Jahn, “The design of a multi-agent transformer condition monitoring system,” Power Systems, IEEE Transactions, vol. 19, no. 4, pp. 1845–1852, 2004.
7. J. Q. Feng, D. P. Buse, Q. H. Wu, and J. Fitch, “A multi-agent based intelligent monitoring system for power transformers in distributed substations,” in Power System Technology, 2002. Proceedings. PowerCon 2002. International Conference, vol. 3, pp. 1962–1965 vol.3, 2002.
8. J. D. McDonald, “Substation automation. IED integration and availability of information,” Power and Energy Magazine, IEEE, vol. 1, no. 2, pp. 22–31, 2003.

9. B. Chaib-draa and F. Dignum, "Trends in agent communication language," *Computational Intelligence*, pp. 89–101, May 2002.
10. F. Bellifemine, G. Caire, and D. Greenwood, *Developing Multi-Agent Systems with JADE*. Wiley Series in Agent Technology, 2004.
11. M. d. Michael Luck, Ronald Ashri, *Agent-Based Software Development*. British Library Cataloguing in Publication Data, 2004.
12. M. Wooldridge, *An Introduction to MultiAgent Systems - Second Edition*. John Wiley & Sons, May 2009.
13. I. A. Ferguson, *TouringMachines: An Architecture for Dynamic, Rational, Mobile Agents* (1992). PhD thesis, University of Cambridge, 1992.
14. M. Elammari and W. Lalonde, "An agent-oriented methodology: High-level and intermediate models," in *Proc. of the 1st Int. Workshop. on Agent-Oriented Information Systems*, 1999.
15. S. A. Deloach, "Analysis and design using mase and agentTool," in *In 12th Midwest Artificial Intelligence and Cognitive Science Conference (MAICS) 2001*, p. 2001.
16. W. Tsaur and H.-C. Tsai, "Secure electronic business applications in mobile agent based networks using elliptic curve cryptosystems," in *Computer Symposium (ICS), 2010 International*, pp. 204–209, 2010.
17. K. Manickavasagam, M. Nithya, K. Priya, J. Shruthi, S. Krishnan, S. Misra, and S. Manikandan, "Control of distributed generator and smart grid using multi-agent system," in *Electrical Energy Systems (ICEES), 2011 1st International Conference*, pp. 212–217, 2011.
18. V. K. Singh and A. K. Gupta, "Agent based models of social systems and collective intelligence," in *Intelligent Agent Multi-Agent Systems, 2009. IAMA 2009. International Conference*, pp. 1–7, 2009.
19. S. D. McArthur, C. D. Booth, J. R. McDonald, and I. T. McFadyen, "An agentbased anomaly detection architecture for condition monitoring," *Power Systems, IEEE Transactions*, vol. 20, no. 4, pp. 1675–1682, 2005.

20. D. P. Buse and Q. H. Wu, "Mobile agents for remote control of distributed systems," *Industrial Electronics, IEEE Transactions*, vol. 51, no. 6, pp. 1142–1149, 2004.
21. F. F. Wu, K. Moslehi, and A. Bose, "Power system control centers: Past, present, and future," *Proceedings of the IEEE*, vol. 93, no. 11, pp. 1890–1908, 2005.
22. G. Diaz, A. Barbon, and J. Gomez-Aleixandre, "Mathematical model of a distribution transformer as applied to turn-to-turn faults analysis," in *EuroPES*, pp. 557–562, June 2002.
23. P. Kang, D. Birtwhistle, J. Daley, and D. McCulloch, "Non-invasive on-line condition monitoring of on load tap changers," in *Power Engineering Society Winter Meeting, IEEE*, vol. 3, pp. 2223–2228, 2000.
24. M. Foata, R. Beauchemin, and C. Rajotte, "On-line testing of on-load tap changers with a portable acoustic system," in *Transmission and Distribution Construction, Operation and Live-Line Maintenance Proceedings. 2000 IEEE ESMO – 2000 IEEE 9th International Conference*, pp. 293–298, 2000.
25. T. Leibfried, "Online monitors keep transformers in service," *Computer Applications in Power, IEEE*, vol. 11, pp. 36–42, Jul 1998.
26. D. Chu and A. Lux, "On-line monitoring of power transformers and components: a review of key parameters," in *Electrical Insulation Conference and Electrical Manufacturing and Coil Winding Conference*, 1999.
27. "IEEE guide for application for monitoring equipment to liquid-immersed transformers and components," *IEEE Std C57.143-2012*, pp. 1–83, 2012.
28. Y. Zhang, X. Ding, "An artificial neural network approach to transformer fault diagnosis," *Power Delivery, IEEE Transactions*, vol. 11, pp. 1836–1841, Oct 1996.
29. S. Mofizul Islam, "A novel fuzzy logic approach to transformer fault diagnosis," *Electrical Insulation, IEEE Transactions*, vol. 7, pp. 177–186, Apr 2000.
30. H. B. Zheng, R. J. Liao, S. Grzybowski, and L. J. Yang, "Fault diagnosis of power transformers using multi-class least square support vector machines classifiers with particle swarm optimisation," *IET Electric Power Applications*, Issue 9, vol. 5, pp. 691–696, 2011.

ДОДАТОК А

Інтелектуальне діагностування стану силового трансформатора із застосуванням мультиагентного підходу

Лістинг програми
УКР.НТУУ “КПІ”.ТВ-з7130мп_18М

Аркушів 17

```

package com.intellectual.diagnostic.controller;
import com.intellectual.diagnostic.entity.Transformer;
import com.intellectual.diagnostic.repo.TransformerRepository;
import com.intellectual.diagnostic.repo.UserRepository;
import com.intellectual.diagnostic.service.ReadDataFromFile;
import com.intellectual.diagnostic.trans.Component;
import com.intellectual.diagnostic.trans.Fault;
import com.intellectual.diagnostic.trans.Symptom;
import com.intellectual.diagnostic.utility.ValidateDataForGraphic;
import com.intellectual.diagnostic.utility.ValidationData;
import lombok.extern.slf4j.Slf4j;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*;
import java.io.IOException;
import java.util.Collection;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
@Controller
@Slf4j
public class MainController {
    @Autowired
    private UserRepository userRepository;
    @Autowired
    private TransformerRepository transformerRepository;
    @GetMapping("/")
    public String greeting() {
        return "index";
    }
    @PostMapping("/technical")
    public String technicalStatePowerTransformer(@RequestParam Double resistanceInsulationWinding,
        @RequestParam Double tangensDielectricalLoss,
        @RequestParam Double windingResistanceDirectCurrent,
        @RequestParam Double lossOfIdlingSpeed,
        @RequestParam Double resistanceShortCircuitWinding,
        Model model) {
        Transformer transformer = Transformer.newBuilder()
            .setResistanceInsulationWinding(resistanceInsulationWinding)
            .setTangensDielectricalLoss(tangensDielectricalLoss)
            .setWindingResistanceDirectCurrent(windingResistanceDirectCurrent)
            .setLossOfIdlingSpeed(lossOfIdlingSpeed)
            .setResistanceShortCircuitWinding(resistanceShortCircuitWinding)
            .build();
        transformerRepository.save(transformer);
        Map<Fault, String> faultStringMap = ValidationData.faultsListTechnicalState(resistanceInsulationWinding,
            tangensDielectricalLoss, windingResistanceDirectCurrent,
            lossOfIdlingSpeed, resistanceShortCircuitWinding);
        Map<Symptom, String> symptomStringMap = ValidationData
            .symptomsListTechnicalState(resistanceInsulationWinding, tangensDielectricalLoss,
            windingResistanceDirectCurrent, lossOfIdlingSpeed, resistanceShortCircuitWinding);
        Map<Component, String> componentStringMap = ValidationData
            .componentsListTechnicalState(resistanceInsulationWinding, tangensDielectricalLoss,
            windingResistanceDirectCurrent, lossOfIdlingSpeed, resistanceShortCircuitWinding);
        int countFaults = ValidateDataForGraphic.countFaults(faultStringMap);
        int countSymptoms = ValidateDataForGraphic.countSymptoms(symptomStringMap);
        int nonDamaged = ValidateDataForGraphic.countFaultsNonDamaged(faultStringMap);
        model.addAttribute("technicalFaults", faultStringMap);
        model.addAttribute("technicalSymptoms", symptomStringMap);
    }
}

```

```

model.addAttribute("technicalComponents", componentStringMap);
log.info("countFaults: " + countFaults);
log.info("countSymptoms: " + countSymptoms);
log.info("nonDamaged: " + nonDamaged);
return "technical";
}
@PostMapping("/chromotographicAnalisis")
public String chromotographicAnalisisGasInOil(@RequestParam Double concentrationOfH2,
        @RequestParam Double concentrationOfCH4,
        @RequestParam Double concentrationOfC2H2,
        @RequestParam Double concentrationOfC2H4,
        @RequestParam Double concentrationOfC2H6,
        @RequestParam Double concentrationOfCO,
        @RequestParam Double concentrationOfCO2,
        @RequestParam Double relativeSpeedBuildUpGas,
        Model model) {
    Transformer transformer = Transformer.newBuilder()
        .setConcentrationOfH2(concentrationOfH2)
        .setConcentrationOfCH4(concentrationOfCH4)
        .setConcentrationOfC2H2(concentrationOfC2H2)
        .setConcentrationOfC2H4(concentrationOfC2H4)
        .setConcentrationOfC2H6(concentrationOfC2H6)
        .setConcentrationOfCO(concentrationOfCO)
        .setConcentrationOfCO2(concentrationOfCO2)
        .setRelativeSpeedBuildUpGas(relativeSpeedBuildUpGas)
        .build();
    transformerRepository.save(transformer);
    Map<Fault, String> faultStringMap = ValidationData
        .faultsListChromotographicAnalisisState(concentrationOfH2, concentrationOfCH4, concentrationOfC2H2,
            concentrationOfC2H4, concentrationOfC2H6, concentrationOfCO, concentrationOfCO2,
            relativeSpeedBuildUpGas);
    Map<Symptom, String> symptomStringMap = ValidationData
        .symptomsListChromotographicAnalisisState(concentrationOfH2, concentrationOfCH4,
            concentrationOfC2H2, concentrationOfC2H4, concentrationOfC2H6, concentrationOfCO, concentrationOfCO2,
            relativeSpeedBuildUpGas);
    Map<Component, String> componentStringMap = ValidationData
        .componentListChromotographicAnalisisState(concentrationOfH2, concentrationOfCH4,
            concentrationOfC2H2, concentrationOfC2H4, concentrationOfC2H6, concentrationOfCO, concentrationOfCO2,
            relativeSpeedBuildUpGas);
    model.addAttribute("chromotographicAnalisisGasFaults", faultStringMap);
    model.addAttribute("chromotographicAnalisisGasSymptoms", symptomStringMap);
    model.addAttribute("chromotographicAnalisisGasComponents", componentStringMap);
    return "chromotographicAnalisis";
}
@PostMapping("/physicChemicalAnalisis")
public String physicChemicalAnalisisOilFromTank(@RequestParam Double breakdownVoltage,
        @RequestParam Double waterContent,
        @RequestParam Double acidNumber,
        @RequestParam Double tangensDielectricalLossOil,
        @RequestParam Double contentAntioxidantAdditive,
        @RequestParam Double contentFuranDerivatives,
        @RequestParam Double gasContent,
        @RequestParam Double contentWaterAcid,
        @RequestParam Double opticalTurbidityOilInTank,
        @RequestParam Double surfaceOilTension,
        @RequestParam Double waterContentHighInsulationTransformer,
        @RequestParam Double resistanceInsulationAvailableRestrainingStraps,
        @RequestParam Double resistanceInsulationYarmoBalka,
        @RequestParam Double contentCopperAndIronInKolloidParts,
        Model model) {

```

```

Transformer transformer = Transformer.newBuilder()
    .setBreakdownVoltage(breakdownVoltage)
    .setWaterContent(waterContent)
    .setAcidNumber(acidNumber)
    .setTangensDielectricalLossOil(tangensDielectricalLossOil)
    .setContentAntioxidantAdditive(contentAntioxidantAdditive)
    .setContentFuranDerivatives(contentFuranDerivatives)
    .setGasContent(gasContent)
    .setContentWaterAcid(contentWaterAcid)
    .setOpticalTurbidityOilInTank(opticalTurbidityOilInTank)
    .setSurfaceOilTension(surfaceOilTension)
    .setWaterContentHighInsulationTransformer(waterContentHighInsulationTransformer)

.setResistanceInsulationAvailableRestrainingStraps(resistanceInsulationAvailableRestrainingStraps)
    .setResistanceInsulationYarmoBalka(resistanceInsulationYarmoBalka)
    .setContentCopperAndIronInKolloidParts(contentCopperAndIronInKolloidParts)
    .build();
    transformerRepository.save(transformer);
    Map<Fault, String> faultStringMap = ValidationData
        .faultsListPhysicChemicalState(breakdownVoltage, waterContent, acidNumber, tangensDielectricalLossOil,
            contentAntioxidantAdditive, contentFuranDerivatives, gasContent, contentWaterAcid,
            opticalTurbidityOilInTank, surfaceOilTension, waterContentHighInsulationTransformer,
            resistanceInsulationAvailableRestrainingStraps, resistanceInsulationYarmoBalka,
            contentCopperAndIronInKolloidParts);
    Map<Symptom, String> symptomStringMap = ValidationData
        .symptomsListPhysicChemicalState(breakdownVoltage, waterContent, acidNumber,
            tangensDielectricalLossOil, contentAntioxidantAdditive, contentFuranDerivatives, gasContent, contentWaterAcid,
            opticalTurbidityOilInTank, surfaceOilTension, waterContentHighInsulationTransformer,
            resistanceInsulationAvailableRestrainingStraps, resistanceInsulationYarmoBalka, contentCopperAndIronInKolloidParts);
    Map<Component, String> componentStringMap = ValidationData
        .componentListPhysicChemicalState(breakdownVoltage, waterContent, acidNumber,
            tangensDielectricalLossOil, contentAntioxidantAdditive, contentFuranDerivatives, gasContent, contentWaterAcid,
            opticalTurbidityOilInTank, surfaceOilTension, waterContentHighInsulationTransformer,
            resistanceInsulationAvailableRestrainingStraps, resistanceInsulationYarmoBalka,
            contentCopperAndIronInKolloidParts);
    model.addAttribute("physicChemicalAnalisisFaults", faultStringMap);
    model.addAttribute("physicChemicalAnalisisSymptoms", symptomStringMap);
    model.addAttribute("physicChemicalAnalisisComponents", componentStringMap);
    return "physicChemicalAnalisis";
}
@PostMapping("/highVoltageInput")
public String controlHighVoltageInputs(@RequestParam Double resistanceInsulation,
    @RequestParam Double tangensDielectricalLossOfTheMainInsulation,
    @RequestParam Double tangensDielectricalLossOfTheLastLayers,
    Model model) {
    Transformer transformer = Transformer.newBuilder()
        .setResistanceInsulation(resistanceInsulation)
        .setTangensDielectricalLossOfTheMainInsulation(tangensDielectricalLossOfTheMainInsulation)
        .setTangensDielectricalLossOfTheLastLayers(tangensDielectricalLossOfTheLastLayers)
        .build();
    transformerRepository.save(transformer);
    Map<Fault, String> faultStringMap = ValidationData.faultsListHighVoltageState(resistanceInsulation,
        tangensDielectricalLossOfTheMainInsulation,
        tangensDielectricalLossOfTheLastLayers);
    Map<Symptom, String> symptomStringMap = ValidationData
        .symptomsListHighVoltageState(resistanceInsulation, tangensDielectricalLossOfTheMainInsulation,
        tangensDielectricalLossOfTheLastLayers);
    Map<Component, String> componentStringMap = ValidationData
        .componentsListHighVoltageState(resistanceInsulation, tangensDielectricalLossOfTheMainInsulation,

```

```

        tangensDielectricalLossOfTheLastLayers);
        model.addAttribute("highVoltageFaults", faultStringMap);
        model.addAttribute("highVoltageSymptoms", symptomStringMap);
        model.addAttribute("highVoltageComponents", componentStringMap);
        return "highVoltageInput";
    }
    @PostMapping("/chromotographicAnalisysGermeticInput")
    public String chromotographicAnalisysInOilGasGermeticInput(@RequestParam Double
gasConcentrationAcetylene,
                                                                @RequestParam Double
sumOfTheHydroCarbonatedGasConcentration,
                                                                @RequestParam Double germeticTurbilityOilInput,
                                                                @RequestParam Double gasContentGermeticInput,
                                                                @RequestParam Double waterContentAcidGermeticInput,
                                                                Model model) {
        Transformer transformer = Transformer.newBuilder()
            .setGasConcentrationAcetylene(gasConcentrationAcetylene)

.setSumOfTheHydroCarbonatedGasConcentration(sumOfTheHydroCarbonatedGasConcentration)
            .setGermeticTurbilityOilInput(germeticTurbilityOilInput)
            .setGasContentGermeticInput(gasContentGermeticInput)
            .setWaterContentAcidGermeticInput(waterContentAcidGermeticInput)
            .build();
        transformerRepository.save(transformer);
        Map<Fault, String> faultStringMap = ValidationData
            .faultsListAnalisysInOilGasGermeticState(gasConcentrationAcetylene,
sumOfTheHydroCarbonatedGasConcentration,germeticTurbilityOilInput,
waterContentAcidGermeticInput);
        Map<Symptom, String> symptomStringMap = ValidationData
            .symptomsListAnalisysInOilGasGermeticState(gasConcentrationAcetylene,
sumOfTheHydroCarbonatedGasConcentration,
germeticTurbilityOilInput,
gasContentGermeticInput,
waterContentAcidGermeticInput);
        Map<Component, String> componentStringMap = ValidationData
            .componentsListAnalisysInOilGasGermeticState(gasConcentrationAcetylene,sumOfTheHydroCarbonatedG
asConcentration, germeticTurbilityOilInput, gasContentGermeticInput,
waterContentAcidGermeticInput);
        model.addAttribute("oilGasGermeticStateFaults", faultStringMap);
        model.addAttribute("oilGasGermeticStateSymptoms", symptomStringMap);
        model.addAttribute("oilGasGermeticStateComponents", componentStringMap);
        return "chromotographicAnalisysGermeticInput";
    }
    @PostMapping("/governingTensionUnderLoad")
    public String devicesGoverningTensionUnderLoad(@RequestParam Double
changerTypePHOAOOn110kV,
                                                                @RequestParam Double changerTypePHOAOOn220kV,
                                                                @RequestParam Double thicknessOfArcingMovingContacts,
                                                                @RequestParam Double effortPressArcingContactsTypePHO,
                                                                @RequestParam Double effortPressMainContacts,
                                                                @RequestParam Double effortPressAdditionalContacts,
                                                                @RequestParam Double thicknessOfVolframContactPasti,
                                                                Model model) throws IOException {
        Transformer transformer = Transformer.newBuilder()
            .setChangerTypePHOAOOn110kV(changerTypePHOAOOn110kV)
            .setChangerTypePHOAOOn220kV(changerTypePHOAOOn220kV)
            .setThicknessOfArcingMovingContacts(thicknessOfArcingMovingContacts)
            .setEffortPressArcingContactsTypePHO(effortPressArcingContactsTypePHO)
            .setEffortPressMainContacts(effortPressMainContacts)
            .setEffortPressAdditionalContacts(effortPressAdditionalContacts)
            .setThicknessOfVolframContactPastie(thicknessOfVolframContactPasti).build();
        transformerRepository.save(transformer);

```



```

        Map<Fault, String> faultStringMap = ValidationData
        .faultsListDevicesGoverningTensionUnderLoad(changerTypePHOAOOn110kV,
changerTypePHOAOOn220kV,      thicknessOfArcingMovingContacts,      effortPressArcingContactsTypePHO,
effortPressMainContacts, effortPressAdditionalContacts, thicknessOfVolframContactPasti);
        Map<Symptom, String> symptomStringMap = ValidationData
        .symptomsListDevicesGoverningTensionUnderLoad(changerTypePHOAOOn110kV,
changerTypePHOAOOn220kV,      thicknessOfArcingMovingContacts,      effortPressArcingContactsTypePHO,
effortPressMainContacts, effortPressAdditionalContacts,thicknessOfVolframContactPasti);
        Map<Component, String> componentStringMap = ValidationData
        .componentsListDevicesGoverningTensionUnderLoad(changerTypePHOAOOn110kV,
changerTypePHOAOOn220kV,thicknessOfArcingMovingContacts,effortPressArcingContactsTypePHO,
effortPressMainContacts, effortPressAdditionalContacts,
        thicknessOfVolframContactPasti);
        model.addAttribute("devicesUnderLoadFaults", faultStringMap);
        model.addAttribute("devicesUnderLoadSymptoms", symptomStringMap);
        model.addAttribute("devicesUnderLoadComponents", componentStringMap);
        return "governingTensionUnderLoad";
    }
    @GetMapping("/technical")
    public String technicalStatePowerTransformer(Model model) {
        model.addAttribute("xxx", "123123");
        return "technical";
    }
    @GetMapping("/chromotographicAnalisys")
    public String chromotographicAnalisysGasInOil(Model model) {
        return "chromotographicAnalisys";
    }
    @GetMapping("/physicChemicalAnalisys")
    public String physicChemicalAnalisysOilFromTank(Model model) {
        return "physicChemicalAnalisys";
    }
    @GetMapping("/highVoltageInput")
    public String controlHighVoltageInputs(Model model) {
        return "highVoltageInput";
    }
    @GetMapping("/chromotographicAnalisysGermeticInput")
    public String chromotographicAnalisysInOilGasGermeticInput(Model model) {
        return "chromotographicAnalisysGermeticInput";
    }
    @GetMapping("/governingTensionUnderLoad")
    public String devicesGoverningTensionUnderLoad(Model model) throws IOException {
        List<Double> governingDataFromFile = ReadDataFromFile.readGoverningTensionUnderLoadData();
        model.addAttribute("governingDataFromFile", governingDataFromFile);
        return "governingTensionUnderLoad";
    }
}

```

```

package com.intellectual.diagnostic.service;
import com.intellectual.diagnostic.entity.TransformerExcelData;
import org.apache.poi.ss.usermodel.*;
import java.io.File;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
import java.util.Objects;
public class ExcelReader {
    public List<TransformerExcelData> readDataFromExcelFile(String path) throws IOException {

```

```

List<TransformerExcelData> transformerExcelDataList = new ArrayList<>();
// Creating a Workbook from an Excel file (.xls or .xlsx)
Workbook workbook = WorkbookFactory.create(new File(path));
Sheet firstSheet = workbook.getSheetAt(0);
Iterator<Row> rowIterator = firstSheet.iterator();
while (rowIterator.hasNext()) {
    Row nextRow = rowIterator.next();
    Iterator<Cell> cellIterator = nextRow.cellIterator();
    TransformerExcelData transformerExcelData = new TransformerExcelData();
    while (cellIterator.hasNext()) {
        Cell nextCell = cellIterator.next();
        int columnIndex = nextCell.getColumnIndex();
        switch (columnIndex) {
            case 1:
                transformerExcelData.setChromotographicAnalysis((Double) getCellValue(nextCell));
                break;
            case 2:
                transformerExcelData.setChromotographicAnalysisGermaticInput((Double)
getCellValue(nextCell));
                break;
            case 3:
                transformerExcelData.setGoverningTensionUnderLoad((Double) getCellValue(nextCell));
                break;
            case 4:
                transformerExcelData.setHighVoltageInput((Double) getCellValue(nextCell));
                break;
            case 5:
                transformerExcelData.setPhysicChemicalAnalysis((Double) getCellValue(nextCell));
                break;
            case 6:
                transformerExcelData.setTechnical((Double) getCellValue(nextCell));
        }
    }
    transformerExcelDataList.add(transformerExcelData);
}
workbook.close();
return transformerExcelDataList;
}

private Object getCellValue(Cell cell) {
    double none = 0;
    switch (cell.getCellType()) {
        case STRING:
            return cell.getStringCellValue();
        case BOOLEAN:
            return cell.getBooleanCellValue();
        case NUMERIC:
            return cell.getNumericCellValue();
        case BLANK:
            cell.setCellValue(0);
            return cell.getNumericCellValue();
    }
    return null;
}

package com.intellectual.diagnostic.service;
import com.intellectual.diagnostic.entity.TransformerExcelData;
import org.apache.poi.ss.usermodel.*;
import java.io.File;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Iterator;

```



```

import java.util.List;
public class ReadDataFromFile {
    private static final String SAMPLE_XLSX_FILE_PATH =
"D:\\some\\path\\Diploma\\src\\main\\resources\\TransformerData.xlsx";
    private static final ExcelReader excelReader = new ExcelReader();
    private static List<TransformerExcelData> listData;
    static {
        try {
            listData = excelReader.readDataFromExcelFile(SAMPLE_XLSX_FILE_PATH);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
    public static List<Double> readChromatographicAnalysisData() throws IOException {
        List<Double> chromatoAnalysisList = new ArrayList<>();
        listData.forEach(x -> chromatoAnalysisList.add(x.getChromotographicAnalysis()));
        return chromatoAnalysisList;
    }
    public static List<Double> readChromatographicAnalysisGermeticInputData() throws IOException {
        List<Double> chromotoGermeticInput = new ArrayList<>();
        listData.forEach(x -> chromotoGermeticInput.add(x.getChromotographicAnalysisGermaticInput()));
        return chromotoGermeticInput;
    }
    public static List<Double> readGoverningTensionUnderLoadData() throws IOException {
        List<Double> governingTension = new ArrayList<>();
        listData.forEach(x -> governingTension.add(x.getGoverningTensionUnderLoad()));
        return governingTension;
    }
    public static List<Double> readHighVoltageData() throws IOException {
        List<Double> hightVoltage = new ArrayList<>();
        listData.forEach(x -> hightVoltage.add(x.getHighVoltageInput()));
        return hightVoltage;
    }
    public static List<Double> readPhysicalChemicalAnalysisData() throws IOException {
        List<Double> physicChemical = new ArrayList<>();
        listData.forEach(x -> physicChemical.add(x.getPhysicChemicalAnalysis()));
        return physicChemical;
    }
    public static List<Double> readTechnicalData() throws IOException {
        List<Double> techical = new ArrayList<>();
        listData.forEach(x -> techical.add(x.getTechnical()));
        return techical;
    }
}

package com.intellectual.diagnostic.entity;
import javax.persistence.*;
import java.io.Serializable;
import java.util.List;
@Entity
public class Transformer implements Serializable {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;
    private Double resistanceInsulationWinding;
    private Double tangensDielectricalLoss;
    //сопротивление обмоток постоянному току
    private Double windingResistanceDirectCurrent;
    private Double lossOfldlingSpeed;
    private Double resistanceShortCircuitWinding;

```

```

private Double concentrationOfH2;
private Double concentrationOfCH4;
private Double concentrationOfC2H2;
private Double concentrationOfC2H4;
private Double concentrationOfC2H6;
private Double concentrationOfCO;
private Double concentrationOfCO2;
private Double relativeSpeedBuildUpGas;
private Double breakdownVoltage;
private Double waterContent;
private Double acidNumber;
private Double tangensDielectricalLossOil;
private Double contentAntioxidantAdditive;
private Double contentFuranDerivatives;
private Double gasContent;
private Double contentWaterAcid;
private Double opticalTurbidityOilInTank;
private Double surfaceOilTension;
private Double waterContentHighInsulationTransformer;
private Double resistanceInsulationAvailableRestrainingStraps;
private Double resistanceInsulationYarmoBalka;
private Double contentCopperAndIronInKolloidParts;
private Double resistanceInsulation;
private Double tangensDielectricalLossOfTheMainInsulation;
private Double tangensDielectricalLossOfTheLastLayers;
private Double gasConcentrationAcetylene;
private Double sumOfTheHydroCarbonatedGasConcentration;
private Double germeticTurbidityOilInput;
private Double gasContentGermeticInput;
private Double waterContentAcidGermeticInput;
private Double changerTypePHOAOn110kV;
private Double changerTypePHOAOn220kV;
private Double thicknessOfArcingMovingContacts;
private Double effortPressArcingContactsTypePHO;
private Double effortPressMainContacts;
private Double effortPressAdditionalContacts;
private Double thicknessOfVolframContactPastie;
private Transformer() {
}
public Long getId() {
    return id;
}
public Double getResistanceInsulationWinding() {
    return resistanceInsulationWinding;
}
public Double getTangensDielectricalLoss() {
    return tangensDielectricalLoss;
}
public Double getWindingResistanceDirectCurrent() {
    return windingResistanceDirectCurrent;
}
public Double getLossOfIdlingSpeed() {
    return lossOfIdlingSpeed;
}
public Double getResistanceShortCircuitWinding() {
    return resistanceShortCircuitWinding;
}
public Double getConcentrationOfH2() {
    return concentrationOfH2;
}

```

```

public Double getConcentrationOfCH4() {
    return concentrationOfCH4;
}
public Double getConcentrationOfC2H2() {
    return concentrationOfC2H2;
}
public Double getConcentrationOfC2H4() {
    return concentrationOfC2H4;
}
public Double getConcentrationOfC2H6() {
    return concentrationOfC2H6;
}
public Double getConcentrationOfCO() {
    return concentrationOfCO;
}
public Double getConcentrationOfCO2() {
    return concentrationOfCO2;
}
public Double getRelativeSpeedBuildUpGas() {
    return relativeSpeedBuildUpGas;
}
public Double getBreakdownVoltage() {
    return breakdownVoltage;
}
public Double getWaterContent() {
    return waterContent;
}
public Double getAcidNumber() {
    return acidNumber;
}
public Double getTangensDielectricalLossOil() {
    return tangensDielectricalLossOil;
}
public Double getContentAntioxidantAdditive() {
    return contentAntioxidantAdditive;
}
public Double getContentFuranDerivatives() {
    return contentFuranDerivatives;
}
public Double getGasContent() {
    return gasContent;
}
public Double getContentWaterAcid() {
    return contentWaterAcid;
}
public Double getOpticalTurbidityOilInTank() {
    return opticalTurbidityOilInTank;
}
public Double getSurfaceOilTension() {
    return surfaceOilTension;
}
public Double getWaterContentHighInsulationTransformer() {
    return waterContentHighInsulationTransformer;
}
public Double getResistanceInsulationAvailableRestrainingStraps() {
    return resistanceInsulationAvailableRestrainingStraps;
}
public Double getResistanceInsulationYarmoBalka() {
    return resistanceInsulationYarmoBalka;
}
}

```

```

public Double getContentCopperAndIronInKolloidParts() {
    return contentCopperAndIronInKolloidParts;
}
public Double getResistanceInsulation() {
    return resistanceInsulation;
}
public Double getTangensDielectricalLossOfTheMainInsulation() {
    return tangensDielectricalLossOfTheMainInsulation;
}
public Double getTangensDielectricalLossOfTheLastLayers() {
    return tangensDielectricalLossOfTheLastLayers;
}
public Double getGasConcentrationAcetylene() {
    return gasConcentrationAcetylene;
}
public Double getSumOfTheHydroCarbonatedGasConcentration() {
    return sumOfTheHydroCarbonatedGasConcentration;
}
public Double getGermeticTurbilityOilInput() {
    return germeticTurbilityOilInput;
}
public Double getGasContentGermeticInput() {
    return gasContentGermeticInput;
}
public Double getWaterContentAcidGermeticInput() {
    return waterContentAcidGermeticInput;
}
public Double getChangerTypePHOAOOn110kV() {
    return changerTypePHOAOOn110kV;
}
public Double getChangerTypePHOAOOn220kV() {
    return changerTypePHOAOOn220kV;
}
public Double getThicknessOfArcingMovingContacts() {
    return thicknessOfArcingMovingContacts;
}
public Double getEffortPressArcingContactsTypePHO() {
    return effortPressArcingContactsTypePHO;
}
public Double getEffortPressMainContacts() {
    return effortPressMainContacts;
}
public Double getEffortPressAdditionalContacts() {
    return effortPressAdditionalContacts;
}
public Double getThicknessOfVolframContactPastie() {
    return thicknessOfVolframContactPastie;
}
public static Builder newBuilder(){
    return new Transformer().new Builder();
}
public class Builder {
    private Builder() {
    }
    public Builder setResistanceInsulationWinding(Double resistanceInsulationWinding) {
        Transformer.this.resistanceInsulationWinding = resistanceInsulationWinding;
        return this;
    }
    public Builder setTangensDielectricalLoss(Double tangensDielectricalLoss) {
        Transformer.this.tangensDielectricalLoss = tangensDielectricalLoss;
    }
}

```

```

    return this;
}
public Builder setWindingResistanceDirectCurrent(Double windingResistanceDirectCurrent) {
    Transformer.this.windingResistanceDirectCurrent = windingResistanceDirectCurrent;
    return this;
}
public Builder setLossOfIdlingSpeed(Double lossOfIdlingSpeed) {
    Transformer.this.lossOfIdlingSpeed = lossOfIdlingSpeed;
    return this;
}
public Builder setResistanseShortCircuitWinding(Double resistanseShortCircuitWinding) {
    Transformer.this.resistanseShortCircuitWinding = resistanseShortCircuitWinding;
    return this;
}
public Builder setConcentrationOfH2(Double concentrationOfH2) {
    Transformer.this.concentrationOfH2 = concentrationOfH2;
    return this;
}
public Builder setConcentrationOfCH4(Double concentrationOfCH4) {
    Transformer.this.concentrationOfCH4 = concentrationOfCH4;
    return this;
}
public Builder setConcentrationOfC2H2(Double concentrationOfC2H2) {
    Transformer.this.concentrationOfC2H2 = concentrationOfC2H2;
    return this;
}
public Builder setConcentrationOfC2H4(Double concentrationOfC2H4) {
    Transformer.this.concentrationOfC2H4 = concentrationOfC2H4;
    return this;
}
public Builder setConcentrationOfC2H6(Double concentrationOfC2H6) {
    Transformer.this.concentrationOfC2H6 = concentrationOfC2H6;
    return this;
}
public Builder setConcentrationOfCO(Double concentrationOfCO) {
    Transformer.this.concentrationOfCO = concentrationOfCO;
    return this;
}
public Builder setConcentrationOfCO2(Double concentrationOfCO2) {
    Transformer.this.concentrationOfCO2 = concentrationOfCO2;
    return this;
}
public Builder setRelativeSpeedBuildUpGas(Double relativeSpeedBuildUpGas) {
    Transformer.this.relativeSpeedBuildUpGas = relativeSpeedBuildUpGas;
    return this;
}
public Builder setBreakdownVoltage(Double breakdownVoltage) {
    Transformer.this.breakdownVoltage = breakdownVoltage;
    return this;
}
public Builder setWaterContent(Double waterContent) {
    Transformer.this.waterContent = waterContent;
    return this;
}
public Builder setAcidNumber(Double acidNumber) {
    Transformer.this.acidNumber = acidNumber;
    return this;
}
public Builder setTangensDielectricalLossOil(Double tangensDielectricalLossOil) {
    Transformer.this.tangensDielectricalLossOil = tangensDielectricalLossOil;
}

```

```

        return this;
    }
    public Builder setContentAntioxidantAdditive(Double contentAntioxidantAdditive) {
        Transformer.this.contentAntioxidantAdditive = contentAntioxidantAdditive;
        return this;
    }
    public Builder setContentFuranDerivatives(Double contentFuranDerivatives) {
        Transformer.this.contentFuranDerivatives = contentFuranDerivatives;
        return this;
    }
    public Builder setGasContent(Double gasContent) {
        Transformer.this.gasContent = gasContent;
        return this;
    }
    public Builder setContentWaterAcid(Double contentWaterAcid) {
        Transformer.this.contentWaterAcid = contentWaterAcid;
        return this;
    }
    public Builder setOpticalTurbidityOilInTank(Double opticalTurbidityOilInTank) {
        Transformer.this.opticalTurbidityOilInTank = opticalTurbidityOilInTank;
        return this;
    }
    public Builder setSurfaceOilTension(Double surfaceOilTension) {
        Transformer.this.surfaceOilTension = surfaceOilTension;
        return this;
    }
}
public Builder setWaterContentHighInsulationTransformer(
Double waterContentHighInsulationTransformer) {
    Transformer.this.waterContentHighInsulationTransformer = waterContentHighInsulationTransformer;
    return this;
}
public Builder setResistanceInsulationAvailableRestrainingStraps(
Double resistanceInsulationAvailableRestrainingStraps) {
    Transformer.this.resistanceInsulationAvailableRestrainingStraps = resistanceInsulationAvailable
RestrainingStraps;
    return this;
}
    public Builder setResistanceInsulationYarmoBalka(Double resistanceInsulationYarmoBalka) {
        Transformer.this.resistanceInsulationYarmoBalka = resistanceInsulationYarmoBalka;
        return this;
    }
}
    public Builder setContentCopperAndIronInKolloidParts(Double contentCopperAndIronInKolloidParts) {
        Transformer.this.contentCopperAndIronInKolloidParts = contentCopperAndIronInKolloidParts;
        return this;
    }
}
    public Builder setResistanceInsulation(Double resistanceInsulation) {
        Transformer.this.resistanceInsulation = resistanceInsulation;
        return this;
    }
}
public Builder setTangensDielectricalLossOfTheMainInsulation(
Double tangensDielectricalLossOfTheMainInsulation) {
    Transformer.this.tangensDielectricalLossOfTheMainInsulation = tangensDielectricalLoss
OfTheMainInsulation;
    return this;
}
}
public Builder setTangensDielectricalLossOfTheLastLayers(
Double tangensDielectricalLossOfTheLastLayers) {
    Transformer.this.tangensDielectricalLossOfTheLastLayers = tangensDielectricalLossOfTheLastLayers;
    return this;
}
}

```

```

    public Builder setGasConcentrationAcetylene(Double gasConcentrationAcetylene) {
        Transformer.this.gasConcentrationAcetylene = gasConcentrationAcetylene;
        return this;
    }
    public Builder setSumOfTheHydroCarbonatedGasConcentration(
        Double sumOfTheHydroCarbonatedGasConcentration) {
        Transformer.this.sumOfTheHydroCarbonatedGasConcentration = sumOfTheHydroCarbonated
        GasConcentration;
        return this;
    }
    public Builder setGermeticTurbidityOilInput(Double germeticTurbidityOilInput) {
        Transformer.this.germeticTurbidityOilInput = germeticTurbidityOilInput;
        return this;
    }
    public Builder setGasContentGermeticInput(Double gasContentGermeticInput) {
        Transformer.this.gasContentGermeticInput = gasContentGermeticInput;
        return this;
    }
    public Builder setWaterContentAcidGermeticInput(Double waterContentAcidGermeticInput) {
        Transformer.this.waterContentAcidGermeticInput = waterContentAcidGermeticInput;
        return this;
    }
    public Builder setChangerTypePHOAOn110kV(Double changerTypePHOAOn110kV) {
        Transformer.this.changerTypePHOAOn110kV = changerTypePHOAOn110kV;
        return this;
    }
    public Builder setChangerTypePHOAOn220kV(Double changerTypePHOAOn220kV) {
        Transformer.this.changerTypePHOAOn220kV = changerTypePHOAOn220kV;
        return this;
    }
    public Builder setThicknessOfArcingMovingContacts(Double thicknessOfArcingMovingContacts) {
        Transformer.this.thicknessOfArcingMovingContacts = thicknessOfArcingMovingContacts;
        return this;
    }
    public Builder setEffortPressArcingContactsTypePHO(Double effortPressArcingContactsTypePHO) {
        Transformer.this.effortPressArcingContactsTypePHO = effortPressArcingContactsTypePHO;
        return this;
    }
    public Builder setEffortPressMainContacts(Double effortPressMainContacts) {
        Transformer.this.effortPressMainContacts = effortPressMainContacts;
        return this;
    }
    public Builder setEffortPressAdditionalContacts(Double effortPressAdditionalContacts) {
        Transformer.this.effortPressAdditionalContacts = effortPressAdditionalContacts;
        return this;
    }
    public Builder setThicknessOfVolframContactPastie(Double thicknessOfVolframContactPastie) {
        Transformer.this.thicknessOfVolframContactPastie = thicknessOfVolframContactPastie;
        return this;
    }
    public Transformer build(){
        return Transformer.this;
    }
}

```

<!DOCTYPE html>


```

<html xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="utf-8">
  <title>Home page</title>
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com
/bootstraps/4.1.3/css/bootstrap.min.css" integrity="sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
crossorigin="anonymous">
  <link th:href="@{/css/style.css}" rel="stylesheet">
</head>
<body>
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="/">Силовий трансформатор</a>
<div class="collapse navbar-collapse" id="navbarSupportedContent">
  <ul class="navbar-nav mr-auto">
    <li class="nav-item active">
      <a class="nav-link" href="/">Головна </a>
    </li>
    <li class="nav-item dropdown">
      <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false">
        Список систем діагностування
      </a>
      <div class="dropdown-menu" aria-labelledby="navbarDropdown">
        <a class="dropdown-item" href="/chromotographicAnalisys">Хроматографічний аналіз
розчинених у маслі газів</a>
        <a class="dropdown-item" href="/chromotographicAnalisysGermeticInput">Хроматографічний
аналіз розчинених в маслі газів герметичних вводів</a>
        <a class="dropdown-item" href="/governingTensionUnderLoad">Прилади регулювання
напруги під навантаженням (ПНН)</a>
        <a class="dropdown-item" href="/highVoltageInput">Контроль високовольтних вводів</a>
        <a class="dropdown-item" href="/physicChemicalAnalisys">Фізико-хімічний аналіз масла
із баку трансформатора</a>
        <a class="dropdown-item" href="/technical">Технічний стан силового трансформатору</a>
      </div>
    </li>
  </ul>
</div>
</nav>
<div class="container">
  <div class="text-center someClass">Діагностування силового трансформатору із застосуванням
мультиагентного підходу </div>
  <div id="carouselExampleIndicators" class="carousel slide" data-ride="carousel">
    <ol class="carousel-indicators">
      <li data-target="#carouselExampleIndicators" data-slide-to="0" class="active"></li>
      <li data-target="#carouselExampleIndicators" data-slide-to="1"></li>
      <li data-target="#carouselExampleIndicators" data-slide-to="2"></li>
    </ol>
    <div class="carousel-inner">
      <div class="carousel-item active">
        
      </div>
      <div class="carousel-item">
        
      </div>
      <div class="carousel-item">
        
      </div>
    </div>
  </div>

```



```

<div class="carousel-item">
  
</div>
<div class="carousel-item">
  
</div>
</div>
<a class="carousel-control-prev" href="#carouselExampleIndicators" role="button" data-
slide="prev">
  <span class="carousel-control-prev-icon" aria-hidden="true"></span>
  <span class="sr-only">Previous</span>
</a>
<a class="carousel-control-next" href="#carouselExampleIndicators" role="button" data-
slide="next">
  <span class="carousel-control-next-icon" aria-hidden="true"></span>
  <span class="sr-only">Next</span>
</a>
</div>
</div>
<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
integrity="sha384-ZMP7rVo3mlykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIpM49"
crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
integrity="sha384-ChfqquxZUCnJSK3+MXmPNlyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"
crossorigin="anonymous"></script>
</body>
</html>

<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="UTF-8">
  <title>Technical information</title>
  <!--<link rel="stylesheet" href="/src/main/webapp/resources/css/style.css">-->
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <!-- Bootstrap CSS -->
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
integrity="sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
crossorigin="anonymous">
</head>
<body>
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="/">Силовий трансформатор</a>
  <div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav mr-auto">
      <li class="nav-item active">
        <a class="nav-link" href="/">Головна</a>
      </li>
      <li class="nav-item dropdown">
        <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-
toggle="dropdown"
aria-haspopup="true" aria-expanded="false">

```

Список систем діагностування

```

</a>
<div class="dropdown-menu" aria-labelledby="navbarDropdown">
  <a class="dropdown-item" href="/chromotographicAnalysis">Хроматографічний аналіз
розчинених у маслі
    газів</a>
  <a class="dropdown-item" href="/chromotographicAnalysisGermeticInput">Хроматографічний
аналіз розчинених в маслі газів герметичних вводів</a>
  <a class="dropdown-item" href="/governingTensionUnderLoad">Прилади регулювання напруги під
навантаженням (РНН)</a>
  <a class="dropdown-item" href="/highVoltageInput">Контроль високовольтних вводів</a>
  <a class="dropdown-item" href="/physicChemicalAnalysis">Фізико-хімічний аналіз масла із баку
трансформатора</a>
  <a class="dropdown-item" href="/technical">Технічний стан силового трансформатору</a>
</div>
</li>
</ul>
</div>
</nav>
<div class="container">
  <div class="text-center">Технічний стан силового трансформатору</div>
  <form method="post">
    <div class="form-group">
      <label for="inputResInsWin">Опір ізоляції обмоток</label>
      <input type="number" name="resistanceInsulationWinding" class="form-control"
id="inputResInsWin"
        placeholder="Введіть опір ізоляції"
        th:value="23"
        required>
    </div>
    <div class="form-group">
      <label for="inputTanDieLos">Тангенс кута діелектричних втрат</label>
      <input type="number" name="tangensDielectricalLoss" class="form-control"
id="inputTanDieLos"
        placeholder="Введіть тангенс кута"
        th:value="47"
        required>
    </div>
    <div class="form-group">
      <label for="inputWinResDirCur">Опір обмоток постійного току</label>
      <input type="number" name="windingResistanceDirectCurrent" step="0.1" class="form-
control"
        id="inputWinResDirCur"
        placeholder="Введіть опір обмоток"
        th:value="2.7"
        required>
    </div>
    <div class="form-group">
      <label for="inputLosIdlSpe">Втрати холостого ходу</label>
      <input type="number" name="lossOfIdlingSpeed" class="form-control" id="inputLosIdlSpe"
        placeholder="Введіть втрати холостого ходу"
        th:value="34"
        required>
    </div>
    <div class="form-group">
      <label for="inputResShoCirWin">Опір короткого замикання обмоток</label>
      <input type="number" name="resistanceShortCircuitWinding" step="0.1" class="form-
control"
        id="inputResShoCirWin"
        placeholder="Введіть опір короткого замикання"

```

```

        th:value="7"
        required>
    </div>
    <button type="submit" class="btn btn-primary">Відправити дані на опрацювання</button>
    <button type="button" class="btn btn-primary" onclick="addCanvas(1,4,3)">Отримати графік
діагностики</button>
</form>
<div class="text-center"><br> Результати діагностування</div>
<div th:if="{not #maps.isEmpty(technicalFaults)}">
    <table>
        <tr>
            <th>Список помилок (несправностей)</th>
        </tr>
        <tr th:each="techFault : ${technicalFaults.values()}">
            <td th:text="{techFault}"></td>
        </tr>
    </table>
</div>
<div th:if="{not #maps.isEmpty(technicalSymptoms)}">
    <table>
        <tr>
            <th>Список симптомів</th>
        </tr>
        <tr th:each="techSymptom : ${technicalSymptoms.values()}">
            <td th:text="{techSymptom}"></td>
        </tr>
    </table>
</div>
<div th:if="{not #maps.isEmpty(technicalComponents)}">
    <table>
        <tr>
            <th>Список пошкоджених компонентів</th>
        </tr>
        <tr th:each="techComponent : ${technicalComponents.values()}">
            <td th:text="{techComponent}"></td>
        </tr>
    </table>
</div>
<div id="sectionForCanvas">
    <canvas id="myChart"></canvas>
</div>
</div>
<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
        integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
        crossorigin="anonymous"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
        integrity="sha384-
ZMP7rVo3mlykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIpM49"
        crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.7.3/Chart.min.js"></script>
<script th:src="@{/js/graphics.js}"></script>
</body>
</html>

```

ДОДАТОК Б

Інтелектуальне діагностування стану силового трансформатора із застосуванням мультиагентного підходу

Акт впровадження
УКР.НТУУ “КПІ”.ТВ-з7130мп_18М

Аркушів 1

2018



АКТ ВПРОВАДЖЕННЯ

Результатів дипломної роботи Поліщука О.В. на тему «Інтелектуальне діагностування стану силового трансформатору із застосуванням мультиагентного підходу», яка виконана в Національному технічному університеті України «Київський політехнічний інститут» ім. І.Сікорського
«06» грудня 2018р.

Нами, представниками кафедри автоматизації проектування енергетичних процесів і систем НТУУ «КПІ» ім. І.Сікорського та Товариством з обмеженою відповідальністю «ВАЙЗ АЙ ТІ», даний акт складено про те, що для використання в розробках спеціалізованого програмного забезпечення Товариством з обмеженою відповідальністю «ВАЙЗ АЙ ТІ», прийняті результати дипломної роботи спеціаліста Поліщука О.В., а саме – програмне забезпечення – система інтелектуального діагностування стану силового трансформатору із застосуванням мультиагентного підходу у вигляді файлу у файлоосховищі, а також документацію програмного супроводу.

Представник кафедри АПЕПС НТУУ «КПІ» ім. І.Сікорського Коваль О.В.

Керівник дипломної роботи

Ковальчук А.М.

Представник Товариства з обмеженою відповідальністю «ВАЙЗ АЙ ТІ»

Удовик Д.В.

Головний спеціаліст

Поліщук О.В.

ДОДАТОК В

Інтелектуальне діагностування стану силового трансформатора із застосуванням мультиагентного підходу

Апробації

УКР.НТУУ “КПІ”.ТВ-з7130мп_18М

Аркушів 5

Л'ОГО

Σ

МИСТЕЦТВО НАУКОВОЇ ДУМКИ

ЗБІРНИК НАУКОВИХ ПРАЦЬ

ЗА МАТЕРІАЛАМИ МІЖНАРОДНОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ

ПІДСУМКИ РОЗВИТКУ НАУКОВОЇ ДУМКИ: 2018

5 ГРУДНЯ 2018 РІК • ІВАНО-ФРАНКІВСЬК, УКРАЇНА

ТОМ 4



ISBN 978-617-7171-80-4

5 грудня 2018 рік • Івано-Франківськ, Україна • 5

ДО ВИЗНАЧЕННЯ ДОВЖИНИ ДУГИ РІЗАННЯ НОЖОМ РОТОРА ПРОСАПНОГО КУЛЬТИВАТОРА Краснолуцький П.П., Кравчук К.О.	65
ДОСЛІДЖЕННЯ СУЧАСНИХ МЕТОДІВ 3D СКАНУВАННЯ Немерцев Д.О.	68
ДОСЛІДЖЕННЯ ПРОЦЕСУ ОБРОБЛЕННЯ ТИТАНОВОГО СПЛАВУ Ti-6Al-4V ЗА ДОПОМОГОЮ ПРОГРАМНОГО ЗАСОБУ DEFORM-3D Рашевський М.Ю.	71
ДОСЛІДЖЕННЯ НАДІЙНОСТІ СИСТЕМИ БУНКЕРУВАННЯ НА ВОДНОМУ ТРАНСПОРТІ ЯК ФАКТОР ЗАБЕЗПЕЧЕННЯ ЕКОЛОГІЧНОЇ БЕЗПЕКИ Мельник О.В., Тимошук О.М.	74
ІНТЕЛЕКТУАЛЬНЕ ДІАГНОСТУВАННЯ СТАНУ СИЛОВОГО ТРАНСФОРМАТОРА ІЗ ЗАСТОСУВАННЯ МУЛЬТИАГЕНТНОГО ПІДХОДУ Ковальчук А.М., Поліщук О.В.	77
ІСТОРІЯ ФРАНЦУЗЬКОЇ КУХНІ Рева А.О., Семко Т.В.	79
КОНВЕНЦІЯ МОП ПРО ПРАЦЮ В РИБАЛЬСЬКОМУ СЕКТОРІ 2007 РОКУ ТА ЇЇ РОЛЬ В ЗАБЕЗПЕЧЕННЯ ПРАЦІ РИБАЛОК Сова О.І.	81
КРИТЕРІЇ ВИБОРУ ЗАСОБІВ ЗАХИСТУ РЕСУРСІВ КОМП'ЮТЕРНОЇ СИСТЕМИ Омельченко Р.В., Захарова М.В.	84
КУПАЖ РОСЛИННИХ ОЛІЙ, ЯК ІНГРЕДІЄНТ ШАМПУНЮ ДЛЯ ВОЛОССЯ Сліпньова В.В.	86
МЕТОД ELECTRE ЯК ІНСТРУМЕНТ ВИБОРУ АЛГОРИТМІВ СИСТЕМИ ФОРМУВАННЯ ПРОПОЗИЦІЙ КОРИСТУВАЧАМ КНИГАРНІ Литвак Р.Б., Дорошенко А.В.	88
МЕТОД РОЗПІЗНАВАННЯ ЖЕСТІВ КАМЕРОЮ ВИКОРИСТАННЯМ МАРКЕРІВ Миценко А.М.	96

ІНТЕЛЕКТУАЛЬНЕ ДІАГНОСТУВАННЯ СТАНУ СИЛОВОГО ТРАНСФОРМАТОРА ІЗ ЗАСТОСУВАННЯМ МУЛЬТИАГЕНТНОГО ПІДХОДУ

А.М. Ковальчук, О.В. Поліщук

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

пр. Перемоги, 37, 03056 Київ, Україна

Силовий трансформатор є важливим компонентом енергетичної системи, який повинен правильно та надійно функціонувати для надійної роботи системи.

Тому інтелектуальне діагностування стану силового дає можливість захистити систему від серйозних проблем, контролювати роботу трансформатору в онлайн режимі.

Під час експлуатації силових трансформаторів виникають різні типи стресів, пов'язаних з навантаженням, температурою навколишнього середовища, часом роботи і т. д. що впливають на його роботу в результаті чого може призвести до втрати електроенергії.

В зв'язку з цим важливо діагностувати помилку якнайшвидше наскільки це можливо, коли виникають певні симптоми. Автоматичні дії можна виконувати навіть в деяких критичних ситуаціях.

Тому для швидкого діагностування потрібно розробити програмне забезпечення для інтелектуального діагностування стану силового трансформатору. Розроблений програмний продукт спрямований на забезпечення діагностування стану трансформатору на основі даних із онлайн моніторингу.

Система призначена для представлення в реальному часі інформації про стан силового трансформатору перед розробкою якої було проведено різноманітні науково-дослідні роботи в області діагностики дефектів силового трансформатора однак декілька проблем як і раніше залишається невирішеним.

Одним з найбільш поширених способів інтелектуального діагностування несправностей у трансформаторі є діагностування на основі онтологій [1].

Однією з основних особливостей застосування онтології є витягнення прихованої інформації з явних фактів, побудованих в онтології. Для розгляду цієї ситуації подано приклад.

Знання в системі знань можуть бути представлені в термінах логіки. У цьому випадку набір понять всередині домену та їх логічні зв'язки між парами понять визначаються як онтологія.

Онтологія забезпечує спільне та загальне розуміння даних, що існують в межах проблемного домену інтеграції програм, та способу полегшення спілкування між людьми та інформаційними системами.

Таким чином, концепція онтології може бути використана для організації та обміну інформацією, управління знаннями та покращення взаємодії систем зв'язку в межах компанії. На основі запропонованої багатоагентної структури онтологія здатна представляти трансформатор та його компоненти.

Онтологія застосовується для діагностики дефектів трансформатора. У цій системі розроблена онтологія, яка відображає взаємозв'язок компонентів трансформатора, ознак несправності та типів несправностей. Мета онтології полягає в тому, щоб забезпечити обмін та повторне використання знань.

Розроблена онтологія діагностики дефектів трансформатора побудована на власній платформі і загорнута розробленим агентом для взаємодії з багатоагентною системою.

Як видно на рисунку 1 структура ієрархії системи побудованої на онтології складається з трьох рівнів; компоненти на найнижчому рівні, агенти знаходяться на другому рівні, і, нарешті, верхній рівень складається з користувачів і систем знань для процесів моніторингу та прийняття рішень, відмічених як моніторинг та рівень прийняття рішень

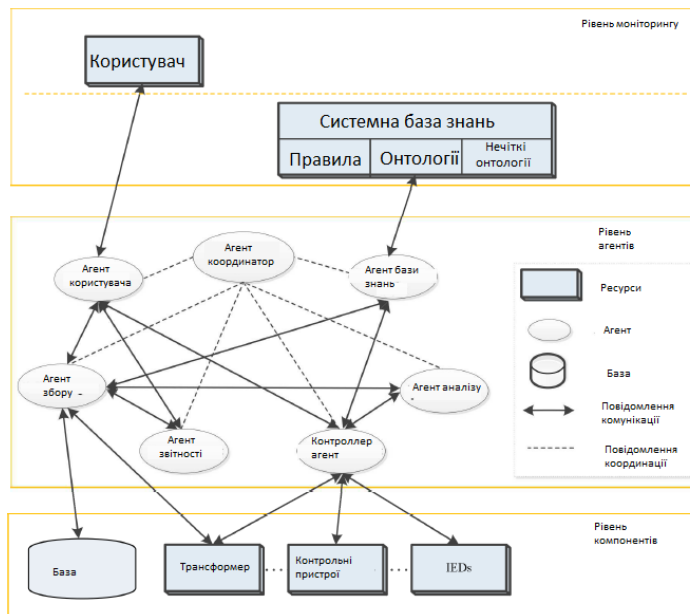


Рис.1. Ієрархія системи інтелектуального діагностування на основі онтологій

Відповідно до даної ієрархії та використовуючи реалізацію мультиагентної системи надається можливість імплементувати програмне забезпечення, яке налагоджує взаємодію між агентами, що попередньо були встановлені у силовому трансформаторі.

Дане програмне забезпечення буди давати користувачу повну глибину оцінки стану силового трансформатору на основі даних отриманих із онлайн моніторингу відповідного силового трансформатору [2].

Тому реалізувавши дану систему у себе на підприємстві де використовується силовий трансформатор, користувачі матимуть змогу передбачити несправності, які можуть з'явитися під час його експлуатації.

СПИСОК ЛІТЕРАТУРИ

1. Попов Г.В. Вопросы диагностики силовых трансформаторов / ФГБОУВПО «Ивановский государственный энергетический университет имени В.И. Ленина». – Иваново, 2012. – 176 с.
2. Тюрюмина А. В., Батрак А. П., Секацкий В. С. Современное состояние вопроса диагностики силовых трансформаторов в зарубежных странах // Молодой ученый. — 2016. — №8. — С. 321-325.